


Article

Chaos-Based Synchronized Dynamic Keys and Their Application to Image Encryption with an Improved AES Algorithm

Chih-Hsueh Lin ¹, Guo-Hsin Hu ^{1,2}, Che-Yu Chan ¹ and Jun-Juh Yan ^{3,*} 

¹ Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung 80778, Taiwan; cslin@nkust.edu.tw (C.-H.L.); guohsin@mail.mirdc.org.tw (G.-H.H.); f1009152146@nkust.edu.tw (C.-Y.C.)

² Department of Industrial Upgrading Service, Metal Industries Research & Development Centre, Kaohsiung 81160, Taiwan

³ Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung 41107, Taiwan

* Correspondence: jjyan@ncut.edu.tw

Abstract: This study aimed to design chaos-based synchronized dynamic keys and develop an improved chaos-based advanced encryption standard (AES) algorithm with the proposed synchronized random keys. First, based on sliding mode control (SMC) technology, a rippling control scheme was introduced to guarantee the synchronization between master–slave discrete chaotic systems. Under the synchronization, the same dynamic random chaos signals could be simultaneously obtained at the transmitter and receiver in communication systems. Then, a novel modified AES cryptosystem with dynamic random keys based on chaos synchronization was presented. In a traditional AES cryptosystem, a static key is used, and it must be exchanged in advance and confirmed to be safely kept. However, in the proposed design, by introducing the synchronization technology of chaotic systems, the static key becomes dynamic and random, and it does not need to be kept or transmitted in open channels. Consequently, the disadvantage of key storage could be eliminated and the security of encryption could be improved. Finally, the developed chaos-based AES (CAES) algorithm has been applied to construct a novel image encryption algorithm. The statistical analysis, histogram, information entropy, and correlation indexes have been calculated and analyzed through simulation experiments in order to demonstrate the capability and improvement of this presented CAES cryptosystem.

Keywords: advanced encryption standard; rippling sliding mode control; synchronization; dynamic random key; image encryption



Citation: Lin, C.-H.; Hu, G.-H.; Chan, C.-Y.; Yan, J.-J. Chaos-Based Synchronized Dynamic Keys and Their Application to Image Encryption with an Improved AES Algorithm. *Appl. Sci.* **2021**, *11*, 1329. <https://doi.org/10.3390/app11031329>

Academic Editor: Ivan A. Parinov

Received: 29 December 2020

Accepted: 27 January 2021

Published: 2 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Since the high-speed growth and wide application in the field of the multimedia information and internet technology, the security demands for data communication and image information confidentiality have continually increased [1]. Recently, because of the good advantages of chaotic signals, such as the broadband random responses, and sensitivity to the system's initial conditions (butterfly effect), many studies have applied these properties to solve and enhance information security and image encryption and decryption issues [2–9]. In the literature [2], a new algorithm that integrated a chaotic system was proposed in order to construct a new transforming-scrambling-diffusion model for color image encryption. In [3], logistic and rectangular chaotic maps were used to implement the image encryption. However, the approaches introduced in the proposed algorithms were simple processes and could not generate uniform distribution for resisting statistical attacks [2]. In [4], the hyper-chaotic system and DNA plane were utilized to propose an image encryption approach. In [5], a new image encryption algorithm was proposed based on complex number

chaotic maps. A hybrid approach combined with a fast chaotic algorithm and advanced encryption standard (AES) image encryption was proposed in the literature [6]. Then, Cramer's rule was used to decompose both encrypted images, and the images could be decrypted using chaos and AES decryption algorithms, respectively. In [7], the logistic map was introduced to construct a chaos cryptosystem, and it was concluded that the logistic map algorithm for image encryption exhibits a good encryption speed. The Arnold chaos system was used to design a round key generator in AES and to combine the chaos random numbers with modified AES in order to propose an algorithm for image encryption in the literature [8]. The advantages of chaotic systems, such as ergodicity and randomness, were also introduced in the literature [9] to ensure a good S-Box for encrypting the plaintext blocks with different S-Box. However, no synchronization procedures were considered in the above reports [2–9]. Therefore, for the chaos-based cryptosystems proposed in these works [2–9], because of the butterfly effect of chaos systems, the transmitter and receiver must have the same initial conditions in order to generate the same chaos random states and complete the correct encryption and decryption. Although the above methods have proposed many analyses and comparisons to prove the security and feasibility of their encryption algorithms, it is under the assumption condition that an unauthorized user does not know the parameters and initial values of their chaotic systems. In other words, the initial conditions play the role of secure keys, and their encryption algorithms are quite secure if the assumption is valid. However, this also implies that when performing encryption and decryption, the above-mentioned parameter information, as well as the initial conditions of the chaotic systems, must be exchanged through a public channel, thereby increasing the possibility of being cracked. Therefore, in this paper, a synchronization technology will be investigated to solve this problem. Such a synchronization design will allow information, such as the parameters and initial conditions of the chaotic system, not be transmitted through a public channel. Through the synchronization technique, only some mixed information, that is not related to system parameters and initial values, needs to be transmitted to the receiving end, and a synchronization controller can be constructed so that the transmitting and receiving ends can simultaneously generate the same chaotic random sequence. In this way, it could be applied to design a new encryption system with high security. In [10], a simple synchronization controller subjected to uncertain chaotic parameters was proposed, and then extended the chaos synchronization to construct secure communication systems. However, this method was applicable for encrypting a large amount of data. By using an independent component analysis, an easy process to derive an effective method for image security was proposed [11]; however, the proposed encryption algorithm can only be applied to square images. Recently, since the concept of chaos synchronization was introduced by Pecora and Carroll [12], research into chaos synchronization and its application in secure communication has become an important issue. Many approaches for controller design have been proposed in order to achieve synchronization for both continuous and discrete chaotic systems, such as backstepping design [13,14], fuzzy control [15,16], and discrete sliding mode control [17]. Among these proposed approaches for control design, the sliding mode control method is frequently used because it has an inherently good robustness, especially for matched disturbances.

On the other hand, the advanced encryption standard (AES) is a well-known symmetric cryptosystem using a static key, and it has been widely realized and fully supported in both hardware and software environments. With a similar structure to AES, there are five different encryption methods, namely AES-CBC, AES-ECB, AES-CFB, AES-OFB, and AES-PCBC [18]. Until now, no available cryptanalytic attacks against AES have been reported. Additionally, AES is also the most commonly used encryption method in Internet of Things (IoT) data transmission encryption [19], because of its high security property. However, in [20], a suite of algorithms was designed to successfully deduce the 128-bit AES key using cache access attacks. Furthermore, in such a symmetric encryption method, a static key was used. When this static key is stolen, information will be fully exposed. In addition, with the recent introduction of IBM's quantum computer with 53 qubits [21], such

an ultra-high computing speed would cause the existing AES encryption to be cracked. Motivated by this, this study aimed to develop a modified chaos-based AES algorithm with dynamic synchronized random keys. Firstly, a control scheme was proposed to solve the problem of synchronization for master–slave chaotic systems. Under synchronization, the same random chaos signals could be simultaneously obtained at the transmitter and receiver in communication systems. Then, the dynamic and unpredictable features of synchronized chaotic signals were extended to design a synchronized dynamic key generator for improving AES cryptosystems. These random keys were dynamically updated and hidden. Unlike the traditional AES algorithm, these keys did not need to be saved in advance, so security could be promoted. To verify the security, the developed chaos-based AES (CAES) algorithm was applied to construct a novel image encryption algorithm. The statistical analysis, histogram, information entropy, and correlation indexes were provided and examined through the simulation experiment in order to demonstrate the efficiency and security of the presented CAES algorithm.

This paper is organized as follows. Section 2 formulates chaos synchronization and the design of a dynamic chaos-based random key generator. The synchronization controller is proposed using discrete sliding mode control. Numerical simulations are given to illustrate the derived results. Section 3 introduces the structure of the improved CAES algorithm. Section 4 proposes and analyzes the performance of the image cryptosystem based on the proposed CAES. Finally, brief conclusions and future works are given in Section 5.

2. Synchronization Design and Dynamic Chaos-Based Random Key Generators

2.1. Chaos Synchronization Design

In this research, a novel CAES cryptosystem was considered with a synchronized chaos-based dynamic random key generator. Before generating the new dynamic chaos-based random keys, the first problem to be well solved was the chaos synchronization. Therefore, the synchronization problem and the controller design were first formulated so as to achieve synchronization for master–slave hyperchaotic Henon maps [22].

Master hyperchaotic Henon map:

$$\begin{aligned}x_1(k+1) &= 1.76 - x_2^2(k) - 0.1 x_3(k) \\x_2(k+1) &= x_1(k) \\x_3(k+1) &= x_2(k)\end{aligned}\tag{1}$$

Slave hyperchaotic Henon map:

$$\begin{aligned}y_1(k+1) &= 1.76 - y_2^2(k) - 0.1y_3(k) + u(k) \\y_2(k+1) &= y_1(k) \\y_3(k+1) &= y_2(k)\end{aligned},\tag{2}$$

where $x = [x_1 \ x_2 \ x_3]^T$ and $y = [y_1 \ y_2 \ y_3]^T$ denote the state vectors of the master and slave chaotic Henon maps, respectively. $u(k)$ is the added controller in the (2) slave system to guarantee the random state vector synchronized with the (1) master system. Strange attractors of the master Henon map (1) are shown in Figure 1. Figure 1 shows some of the characteristics of chaotic systems, such as strange attractors and unpredictability to the random-like signal response.

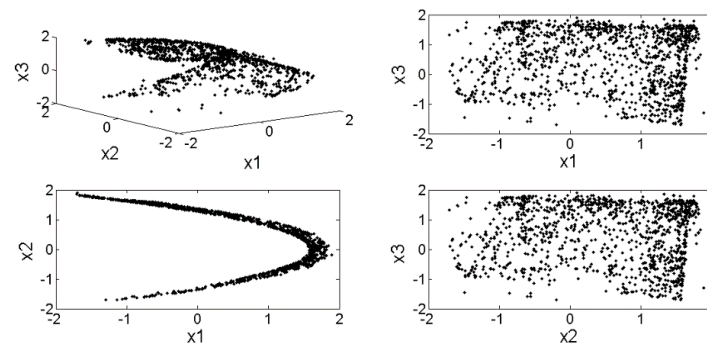


Figure 1. Strange attractors of the master Henon map.

To discuss the synchronization controller design, the error state was defined as $e_i(k) = y_i(k) - x_i(k), i = 1, 2, 3$. This resulted in the following error system:

$$\begin{aligned} e_1(k+1) &= -e_2(k)(x_2(k) + y_2(k)) - 0.1e_3(k) + u(k) \\ e_2(k+1) &= e_1(k) \\ e_3(k+1) &= e_2(k) \end{aligned} \quad (3)$$

Here, the aim was to design a controller, $u(k)$, to result in $\lim_{k \rightarrow \infty} \|e_i(k)\| = 0, i = 1, 2, 3$. In other words, the states of master–slave hyperchaotic systems described in Equations (1) and (2) could be fully synchronized with random chaos dynamics. To obtain a good robustness, similar to that of the literature [23], the sliding mode control was introduced to design the controller ($u(k)$) in order to ensure the exact synchronization. Generally, there are two steps for completing the design of the sliding mode controller ($u(k)$). First, it is necessary to design a proper sliding surface ($s(k)$) and to ensure the state synchronization of the controlled system under this assumption, such that the controlled trajectory of the systems can hit and operate in the sliding mode (i.e., $s(k) = 0$). Then, it is necessary to design the sliding mode controller ($u(k)$) to guarantee that the state trajectory of the controlled states can be forced to the sliding manifold as expected. In order to complete the above design steps, the sliding surface for the sliding manifold was selected as follows:

$$s(k) = e_1(k) + \alpha e_2(k) + \beta e_3(k) \quad (4)$$

where $s(k) \in R$ and α, β are the chosen parameters. If the controlled system can be forced to enter the sliding motion under the effect of $u(k)$ (in other words, $s(k) = 0$), then $e_1(k) = -\alpha e_2(k) - \beta e_3(k)$. By substituting the relation of $e_1(k) = -\alpha e_2(k) - \beta e_3(k)$ into the error Equation (3) of the master–slave system, it can be rewritten as follows:

$$\begin{bmatrix} e_2(k+1) \\ e_3(k+1) \end{bmatrix} = \begin{bmatrix} -\alpha & -\beta \\ 1 & 0 \end{bmatrix} \begin{bmatrix} e_2(k) \\ e_3(k) \end{bmatrix} = AE(k) \quad (5)$$

From Equation (5), obviously, if α and β are chosen to ensure $|\lambda_i(A)| < 1, i = 1, 2$, then $E(k) = [e_2(k) \ e_3(k)]^T$ can converge to zero. Furthermore, as the controlled systems operate in the sliding mode, $e_1(k) = -\alpha e_2(k) - \beta e_3(k)$ and $e_1(k) = 0$ are ensured when $E(k) = 0$. Therefore, the state responses of the controlled slave Henon map (2) can fully track the random chaotic behavior of master Henon map (1) in the sliding manifold ($s(k) = 0$).

After discussing the synchronization under the condition of $s(k) = 0$, the sliding mode controller $u(k)$ is given as follows:

$$u(k) = -[\alpha e_1(k) + (\beta - x_2(k) - y_2(k))e_2(k) - 0.1e_3(k)] + \delta s(k) \quad (6)$$

where $|\delta| < 1$. If the sliding mode controller is given as Equation (6), the following equation can be obtained:

$$\begin{aligned} s(k+1) &= \alpha e_1(k) + (\beta - x_2(k) - y_2(k))e_2(k) - 0.1e_3(k) + u(k) \\ &= \delta s(k) \end{aligned} \quad (7)$$

According to Equation (7), $s(k+1) = \delta s(k)$. Because $|\delta| < 1$ was specified, the trajectory could be forced to hit and operate in the sliding manifold ($s(k) = 0$). In addition, the state behavior of the master–slave Henon maps would be synchronized and would have the same chaotic dynamics as in Figure 1.

Next, the simulation tool of MATLAB was used to examine the control design discussed above. In this simulation, the initial conditions were selected as $x(0) = [1.5 \ -0.3 \ 0.4]^T$ and $y(0) = [-0.3 \ -1.1 \ 0.8]^T$. $\alpha = 0.1$ and $\beta = 0.3$ were selected, such that the eigenvalues of A were $\lambda_1 = 0.6$, $\lambda_2 = -0.5$, and $\delta = 0.5$. Therefore, the sliding surface could be constructed as follows:

$$s(k) = e_1(k) + 0.1e_2(k) + 0.3e_3(k) \quad (8)$$

According to Equation (6), the controller was designed as follows:

$$u(k) = -[0.1e_1(k) + (0.3 - x_2(k) - y_2(k))e_2(k) - 0.1e_3(k)] + 0.5s(k) \quad (9)$$

Figures 2–4 show the simulation results. Figure 2 shows the error state with the proposed control input (Equation (9)). Figure 3 shows the state responses of the controlled systems. The responses of the switching function of $s(k)$ and the sliding mode controller $u(k)$ were, respectively, given in Figure 4. By observing the simulation results, it was revealed that the trajectory of the controlled systems could be forced to the sliding manifold of $s(k) = 0$; the error state responses between the master–slave chaotic system converged to zero, as expected; and the controlled master–slave system reached exact synchronization.

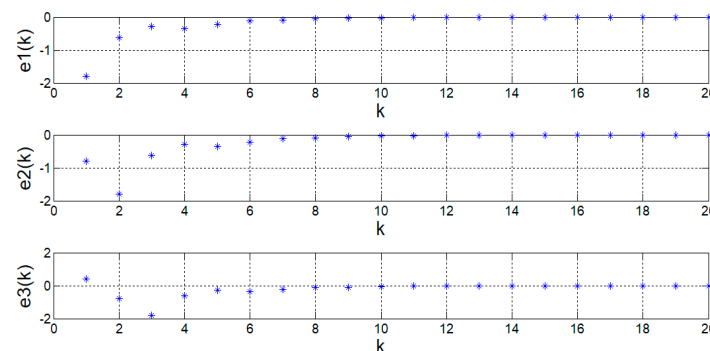


Figure 2. Time responses of error states.

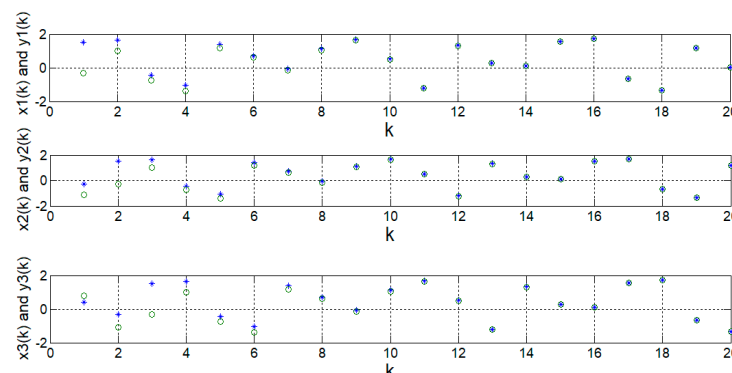


Figure 3. Time responses of controlled states.

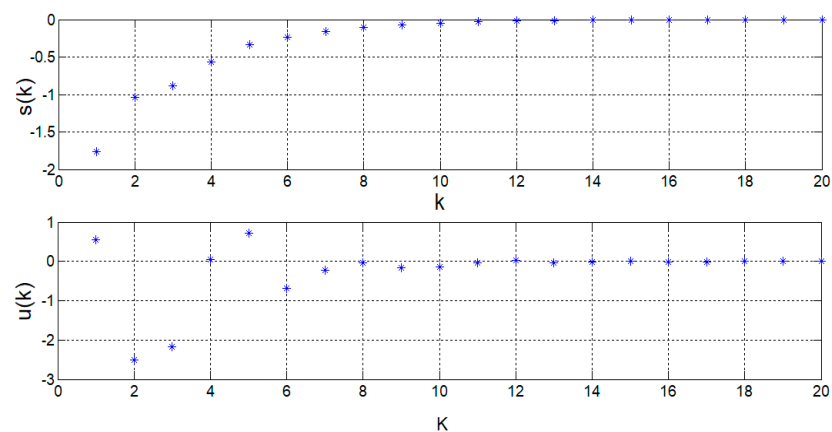


Figure 4. Time responses of the switching surface ($s(k)$) and controller ($u(k)$).

2.2. Dynamic Chaos-Based Random Key Generator

From the above discussion, it was possible to suppress the butterfly effect between master–slave systems through the effect of the controller, and the states of the controlled slave system could be synchronized with those random states of the master system. Next, we designed a dynamic random key generator. There are three types of AES advanced encryption technologies, which use cryptographic keys of 128, 192, and 256 bits [24], respectively. Obviously, from the above chaotic systems, it was possible to generate three random states; however, if these random states were used as the AES encryption key, the lengths would be somewhat insufficient. Therefore, to solve this problem, dynamic chaotic signals were integrated with the hash function to establish the dynamic key generator. Because of the avalanche effect of Secure Hash Algorithm 256 (SHA256) [25] and the butterfly effect of the chaotic systems, each dynamic key was never repeated, which made frequency analysis impossible and thus was difficult to crack. The dynamic random key generator with SHA256 is shown in Figure 5.

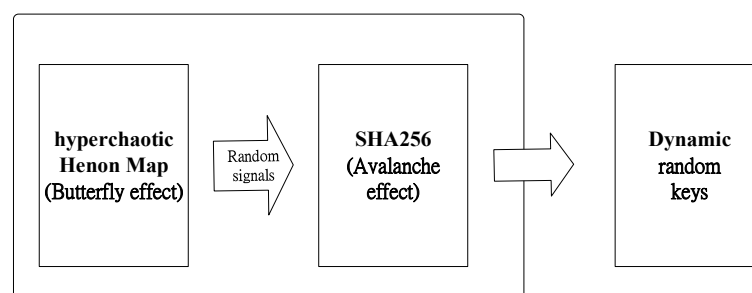


Figure 5. The dynamic random key generator with SHA256.

3. Design of the Improved AES Algorithm

AES encryption is a popular symmetric-key cryptosystem. The main architecture of the AES encryption algorithm is given in Figure 6. In order to generate a uniform distribution of cipher text, a series of linked operations were included in AES cryptosystem design. Some of the linked operations were performed by substituting the inputs using some output feedback (substitutions). The left operations involved shuffling many bits around (permutations). AES runs computations using bytes. Therefore, the AES separated 128 bits of information block into 16 bytes. These 16 bytes were then formatted as a matrix form with four columns and four rows in order to process the operations. The operation rounds in the AES cryptosystem were dependent on the key length we selected. Here, 10 rounds, 12 rounds, and 14 rounds, were used for the key lengths of 128-bit, 192-bit, and 256-bit, respectively, as shown in Figure 6. Using different round keys, K_R , derived from the original static AES key, each of these rounds was performed, where R was the round.

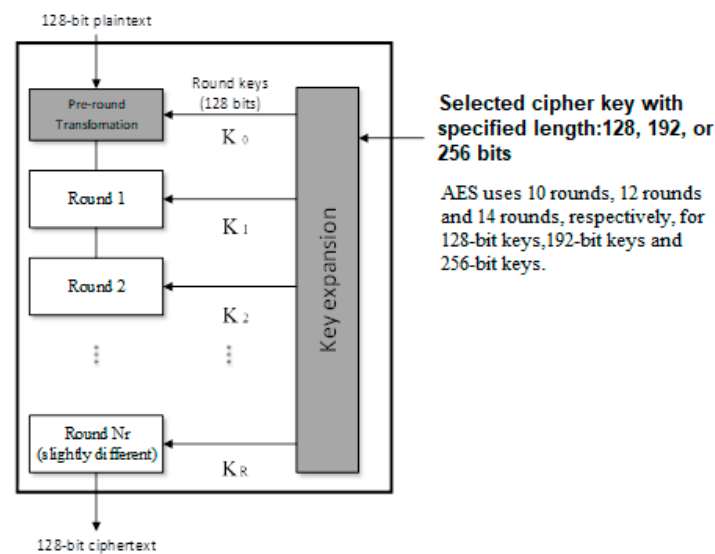


Figure 6. Advanced encryption standard (AES) architecture.

For AES symmetric-key cryptosystems, the keys represented a shared secret in the communication systems that could guarantee private information transmission. However, compared with asymmetric key encryption, one of the main weaknesses of symmetric key cryptosystems was the requirement of the secret key [25]. Once this static secret key is stolen, all information will be exposed. Therefore, to solve this problem, in this paper, a new CAES encryption method was proposed based on synchronized chaotic dynamic keys, as shown in Figure 7.

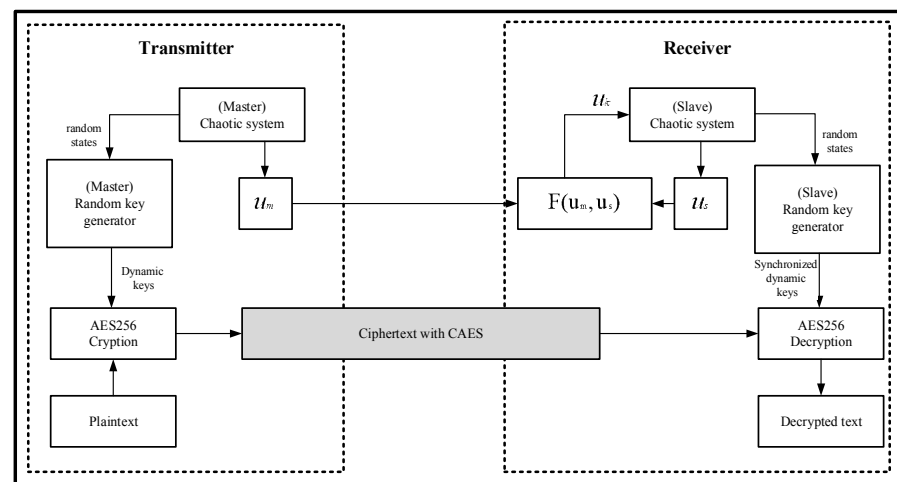


Figure 7. Improved chaos-based AES architecture.

In this proposed improved chaos-based AES architecture, the dynamic random states of synchronized chaos systems were used as the secure keys. Obviously, this improved design could provide synchronized random states to a random key generator and then generate dynamic random keys with 256 bits for AES to promote its security. By introducing a designed controller to the slave chaotic system, the slave states could be synchronized with the master states. Furthermore, to promote the security of practical applications, the synchronization controller ($u(k)$) was decomposed into $u_m(k)$ and $u_s(k)$, calculated from the master and slave chaotic systems, respectively. The controller was then realized at the receiver side with $u(k) = F(u_m(k), u_s(k))$. Even if hackers had these parameters in the synchronization controller, they would still not be able to obtain the states of the master and slave systems separately, and they would not be able to reorganize the synchronization

controller. Therefore, the security of the dynamic keys and the encryption system would be ensured. After achieving synchronization, the same random chaos signals could be simultaneously obtained, and the synchronized dynamic random keys at both sides of transmitter and receiver could be constructed. The static keys in the traditional AES were replaced by synchronized random signals, and did not need to be given in advance or be communicated through public channels. As a result, the disadvantage of key storage could be eliminated and the security of the encryption could be improved because of the synchronized dynamic random keys.

4. Implementation and Performance Analysis of CAES-Based Image Encryption

After designing the CAES algorithm shown in Figure 7, this proposed CAES was applied to image encryption, and its encryption performance and security were analyzed. Here, Python was used as the developing environment. The program modules used included Pycrypto, Numpy, Numpydoc, OpenCV, and Matplotlib. In this image encryption, PNG images were used as the plain text data. The encrypted files could also be displayed normally by pixel processing. At the same time, the encrypted pixel data were stored in bytes as the test data for the National Institute of Standards and Technology (NIST) test. In this application, the master–slave hyperchaotic Henon Maps in Equations (1) and (2) were used with a synchronization controller from Equation (6) to synchronize dynamic random key generators with SHA256, as shown in Figure 7. The encryption and decryption of the CAES flow chart are shown in Figures 8 and 9, respectively.

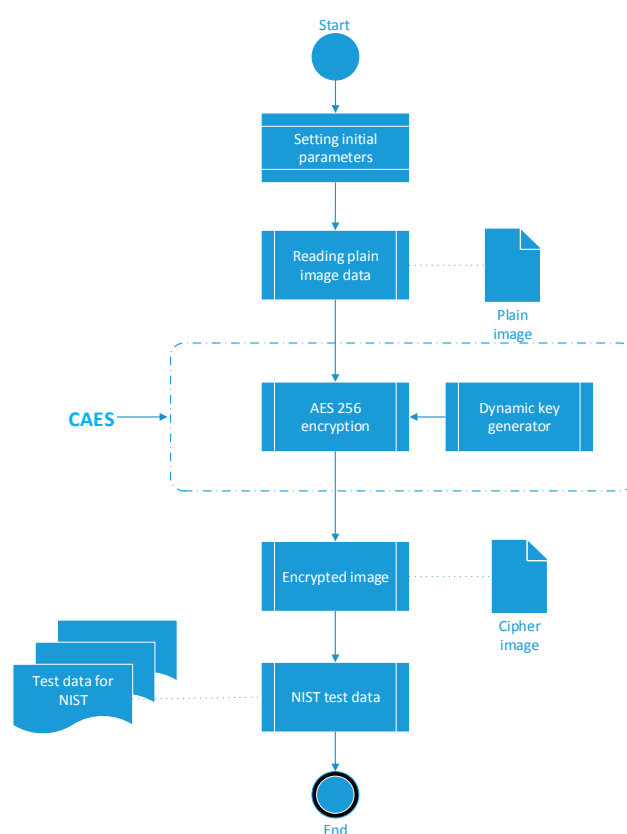


Figure 8. Flow chart of encryption using chaos-based AES (CAES).

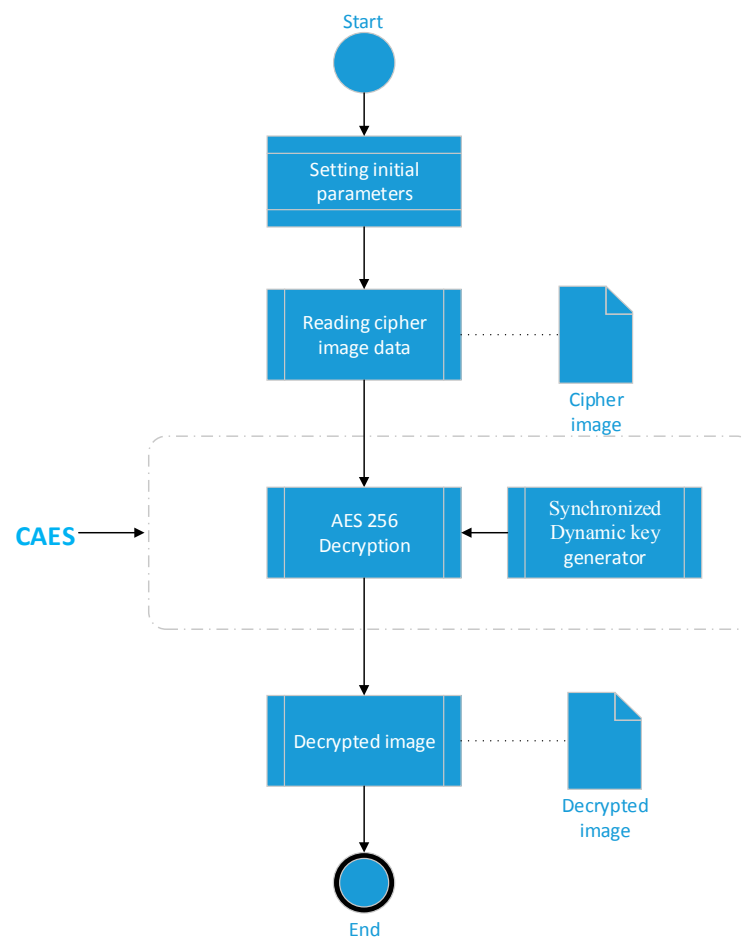


Figure 9. Flow chart of decryption using CAES.

4.1. Visual Effect of Encrypted Images

In this example, an image with a 256×256 resolution was used as the plaintext data. Several of the plain and cipher images are shown below.

From Figure 10, it can be observed that when the AES-ECB was used to encrypt the data, the intervals with the same color in the background would be encrypted into the same cipher text. This was a drawback of AES-ECB for image encryption. The encrypted picture looked confusing enough when AES-CBC was used to encrypt the image. When using the proposed improved CAES for pictures with a size of 256×256 and 49,152 dynamic keys, as the original AES already had a qualified security, it would be impossible to completely solve all of the keys in short time. Furthermore, because of the design of the dynamic key, the 49,152 keys were updated immediately in the next round of encryption and the old password had no effect. Therefore, in terms of security, CAES was greatly improved compared with the original AES-ECB and AES-CBC.

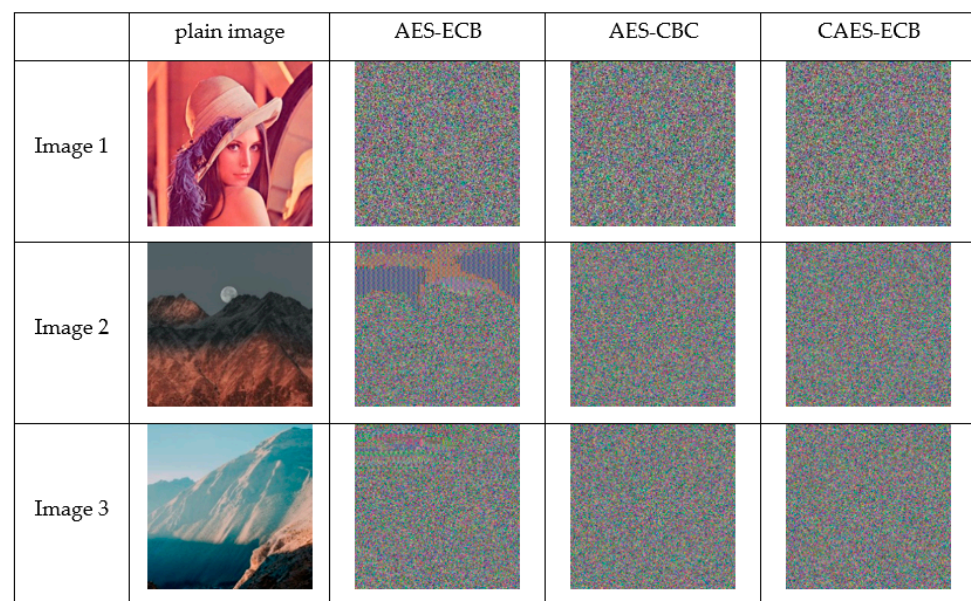


Figure 10. Visual effect of encrypted images.

4.2. Statistical Analysis by NIST

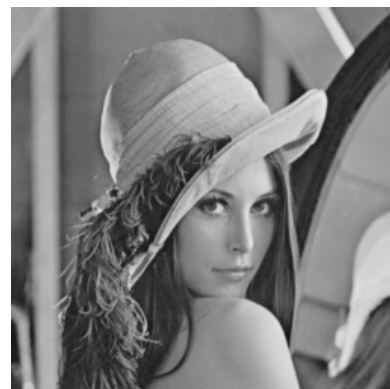
In this section, the National Institute of Standards and Technology (NIST) [26] test suite was introduced to evaluate the randomness of the encrypted images. The OpenCV function library of Python was utilized to collect the three values of R, G, and B from image 1 in Figure 10; then convert the values from 0 to 255 into bytes for encryption, as shown in Figure 8; and then output the results as a NIST test file through the Numpy library. The NIST SP800-22 test consisted of 15 items. The method of detection compared the calculated outputted p -value for each item with a prescribed significance level of 0.01. If the outputted p -values were all greater than 0.01 for all of the test items, it passed the NIST test. The NIST test results shown in Table 1 were obtained with a stream length of 10,000,000 bits and a bit stream count of 30. The obtained data revealed that the encrypted data from the CAES algorithm passed the NIST SP 800-22 random number detection (p -value > 0.001) and that the test performance of each item was better than the original AES-CBC and AES-ECB. This result indicates that the presented CAES algorithm had a good randomness.

Table 1. National Institute of Standards and Technology (NIST) randomness test.

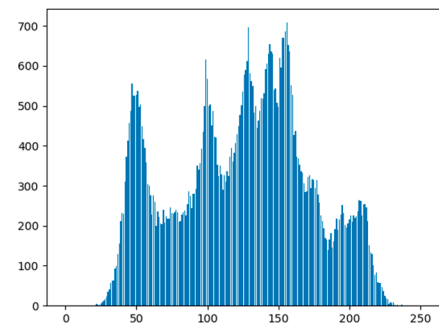
Nist Test	p -Value			
	AES-ECB	AES-CBC	CAES-ECB	CAES-CBC
Frequency	0.000000	0.468595	0.602458	0.862344
Block frequency	0.000954	0.534146	0.602458	0.350485
Cumulative sums	0.000001	0.949602	0.602458	0.999896
Runs	0.000000	0.407091	0.991468	0.299251
Longest run	0.000000	0.534146	0.739918	0.739918
Rank	0.000000	0.534146	0.350485	0.407091
FFT	0.000000	0.468595	0.407091	0.253551
Non overlapping template	0.000000	0.998205	0.991468	0.991468
Overlapping template	0.000439	0.000737	0.407091	0.911413
Universal	0.000000	0.804337	0.862344	0.862344
Approximate entropy	0.000000	0.407091	0.862344	0.299251
Random excursions	0.122325	0.484646	0.350485	0.534146
Random excursions variant	0.025193	0.980883	0.637119	0.911413
Serial	0.000000	0.350485	0.911413	0.804337
Linear complexity	0.000000	0.299251	0.299251	0.671779
Sum	0.148912	8.221956	9.617851	9.898687
Average	0.009927	0.548130	0.641190	0.659912

4.3. Histogram Analysis and Mean Absolute Deviation

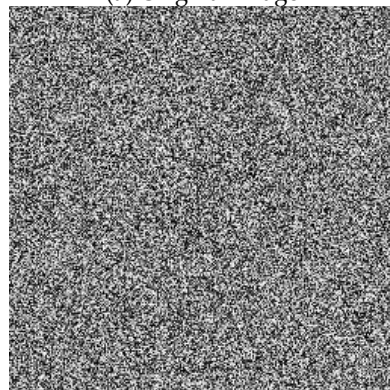
For an image, the histogram is an important statistical feature. A good cryptosystem can encrypt an image, and the histogram of the cipher image is evenly distributed, such that attackers can not analyze the statistical features of the original image from its cipher image. Figure 11a–d presents the visual effect of the plain and cipher images, respectively. Figure 11e–h shows the histograms of the plain and cipher images, respectively, demonstrating that the encryption can safely obscure the histogram information.



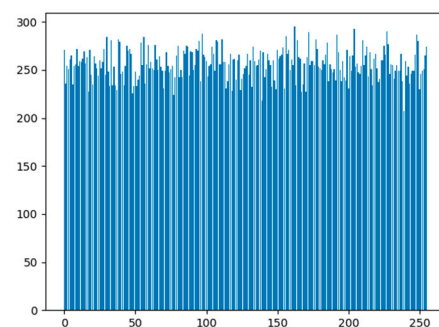
(a) Original image



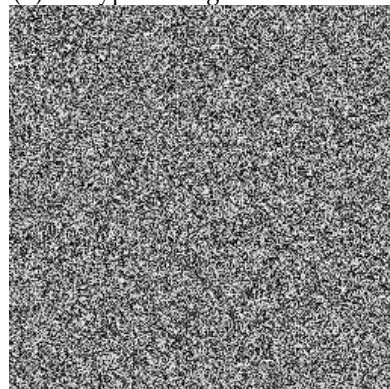
(e) Histogram of original image



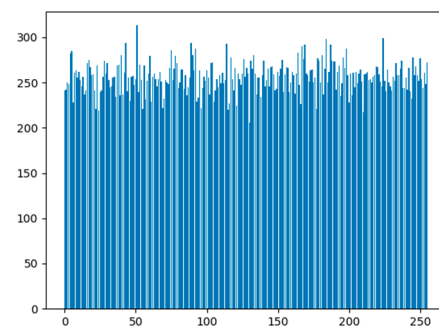
(b) Encrypted image with AES-ECB



(f) Histogram of encrypted image with AES-ECB

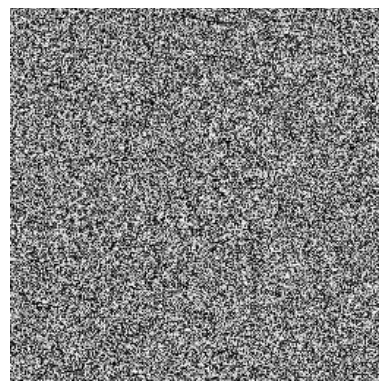


(c) Encrypted image with AES-CBC

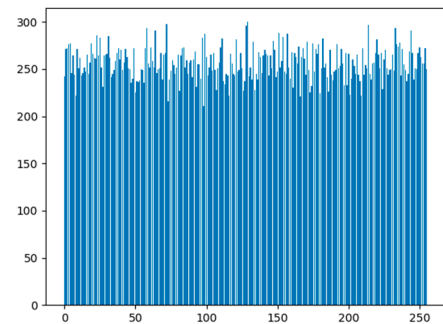


(g) Histogram of encrypted image with AES-CBC

Figure 11. *Cont.*



(d) Encrypted image with CAES-CBC



(h) Histogram of encrypted image with CAES-CBC

Figure 11. The visual effect and histograms of the plain and cipher images.

To evaluate the uniform distribution of histograms, the mean absolute deviation (MAD) [27] was calculated using the following equation:

$$MAD = \frac{1}{256} \sum_{i=0}^{255} |x_i - m(X)|, \quad (10)$$

where x_i is the number of pixel values (i), $X = \{x_0, x_1, \dots, x_{255}\}$ and $m(X) = \frac{1}{256} \sum_{i=0}^{255} x_i$ is the mean of X . Obviously, for images with a size of 256×256 , $m(X) = 256$.

From Table 2, it is revealed that not all of the histograms of the encrypted images from AES-EBC had a uniform distribution. However, AES-EBC and CAES-CBC always had good mean absolute deviations. Furthermore, CAES-CBC had the smallest average MAD. This means that the uniform distribution of CAES-CBC was slightly better than AES-CBC.

Table 2. Mean absolute deviation (MAD) uniform distribution test.

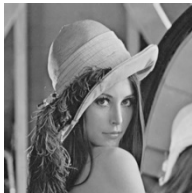

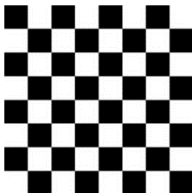

Image	Histogram MAD Uniform Distribution Test		
	AES-ECB	AES-CBC	CAES-CBC
	12.3	12.67	12.67
	33.56	12.68	12.99
	93.51	12.32	12.07

Table 2. Cont.

Image	Histogram MAD Uniform Distribution Test		
	AES-ECB	AES-CBC	CAES-CBC
	13.93	12.41	11.87
Average MAD	38.325	12.52	12.4

4.4. Information Entropy Analysis

Image information entropy (IE) represents the aggregation feature of the distribution of the image pixel values [28]. The image IE was calculated using the following formula:

$$H = - \sum_{i=1}^{255} p_i \log p_i \quad (11)$$

where p_i is the frequency of each greyscale. For a grayscale image, the pixel had a data field of [0, 255], and the maximum value of IE will be 8. The IE test report with the grayscale of image 1 in Figure 10 is given in Figure 12. From Figure 12, the IE value of CAES-ECB is the closest to the ideal value of 8. Therefore, it can be concluded that the encrypted image with CAES-ECB possessed the true random signal property, and the proposed algorithm had the ability to resist entropy attacks.

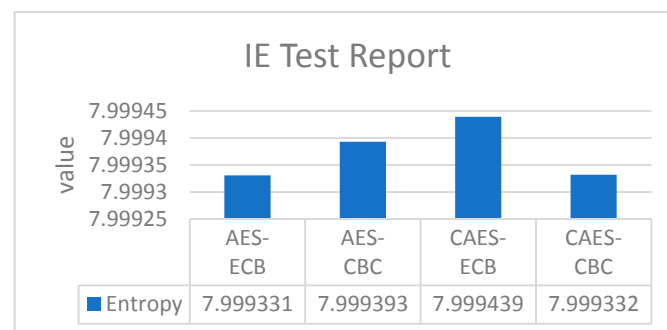


Figure 12. Information entropies for comparison.

4.5. Connected Component Analysis

Generally speaking, for a plain image, a high correlation between adjacent pixels is an obvious feature. Therefore, it would be better if the encryption algorithm could result in smaller correlation values between adjacent pixels. Here, 3000 pixels were randomly selected from the information of the plaintext and encrypted images of image 1 in Figure 10. The correlation coefficients for the horizontal, vertical, and diagonal directions were calculated using the following equations [2]:

$$CCA = \frac{\sum_{i=1}^N (x_i - \frac{1}{N} \sum_{i=1}^N x_i)(y_i - \frac{1}{N} \sum_{i=1}^N y_i)}{\sqrt{\sum_{i=1}^N (x_i - \frac{1}{N} \sum_{i=1}^N x_i)^2 \times \sum_{i=1}^N (y_i - \frac{1}{N} \sum_{i=1}^N y_i)^2}} \quad (12)$$

where x_i and y_i denote the pixel values of the two adjacent pixels, and N is the number of pair (x_i, y_i) .

Figure 13 shows the correlation in a diagonal direction for the plain and encrypted image information according to CAES-ECB, separately. A detailed report of the connected component analysis is given in Figure 14. According to Figure 14, it can be seen that AES-ECB had the highest pixel correlation. In AES-CBC, although it was superior to AES-ECB, the correlation of the vertical and horizontal directions was higher than those of the proposed CAES-ECB. Although AES-CBC had a lower pixel correlation in the diagonal direction, the proposed CAES-ECB still had the lowest average correlation. This result indicates that the proposed improved CAES algorithm had better diffusion characteristics than AES-ECB and AES-CBC.

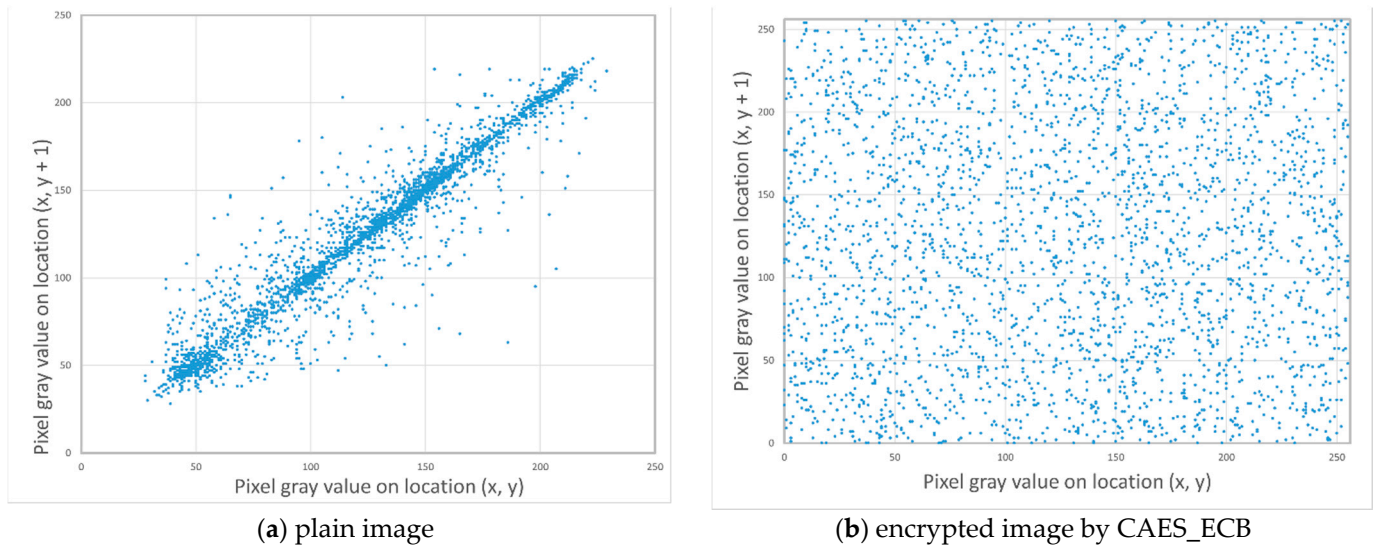


Figure 13. Correlation of the diagonal direction.

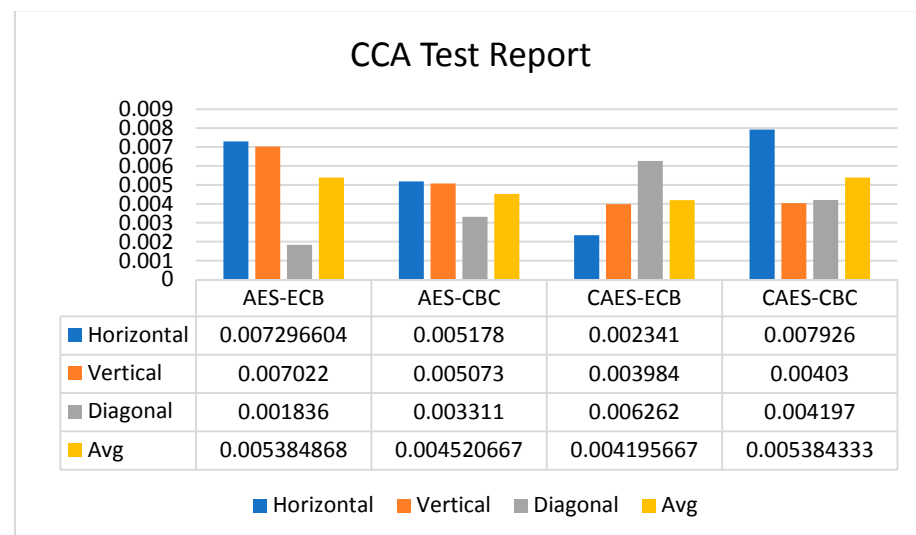


Figure 14. Test report of the connected component analysis.

5. Conclusions and Future Works

In this paper, by integrating the random property of chaotic signals with synchronization control, the synchronized random key generator was established. Then, a novel modified CAES cryptosystem with the proposed dynamic random keys was presented. Unlike the traditional AES algorithm, the traditional AES static key was replaced by the proposed synchronized dynamic random keys, not necessary to be saved or exchanged in advance, so that the security could be prompted. This proposed CAES was applied to

image encryption, and several tests and analyses, such as visual effect, statistical analysis, histogram, information entropy analysis, and connected component analysis, were all performed to determine the capability and security of the CAES algorithm. Recently, as a result of the impact of COVID-19 and to reduce the risk of infection caused by conversation contact, people have become accustomed to using wireless networks for communication. In addition, with the ultra-high computing speed of quantum computers, the existing encryption algorithms might be able to be cracked. Therefore, knowing how to ensure the security of data exchange is a very important issue. In the future, the robustness and synchronization speed of the proposed random key generators should be further discussed and promoted, such that the chaos-based cryptosystem could be applied in order to ensure the communication security of video/audio streaming in the network environment.

Author Contributions: Conceptualization, C.-H.L., G.-H.H., C.-Y.C., and J.-J.Y.; methodology, C.-H.L., G.-H.H., and J.-J.Y.; software, G.-H.H. and C.-Y.C.; validation, G.-H.H. and C.-Y.C.; formal analysis, C.-H.L. and J.-J.Y.; investigation, C.-H.L. and G.-H.H.; writing—original draft preparation, C.-H.L. and J.-J.Y.; writing—review and editing, J.-J.Y.; visualization, G.-H.H. and C.-Y.C.; supervision, C.-H.L.; project administration, J.-J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the Ministry of Science and Technology, Taiwan, under MOST-107-2221-E-167-032-MY2 and MOST-109-2221-E-992-070.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Durdi, V.B.; Kulkarni, P.T.; Sudha, K.L. Selective encryption framework for secure multimedia transmission over wireless multimedia sensor networks. *Advances in Intelligent Systems and Computing*. 2017. In *Proceedings of the International Conference on Data Engineering and Communication Technology*, Lavasa, Pune, Maharashtra, India, 10–11 March 2016; Springer: Singapore, 2017; Volume 469.
2. Li, C.; Zhao, F.; Lei, C.; Lei, L.; Zhang, J. A hyperchaotic color image encryption algorithm and security analysis. *Secur. Commun. Netw.* **2019**, 2019. [[CrossRef](#)]
3. Zhang, Y.H.; Zhang, B. Algorithm of image encrypting based on Logistic chaotic system. *Appl. Res. Comput.* **2015**, 32, 1770–1773.
4. Liu, Y.; Lin, T.; Wang, J.; Yuan, H.M. An image encryption algorithm based on hyper-chaos system and DNA plane. *J. Comput.* **2018**, 29, 43–55.
5. Liu, Y.; Tong, X.; Hu, S. A family of new complex number chaotic maps based image encryption algorithm. *Signal Process. Image Commun.* **2013**, 28, 1548–1559. [[CrossRef](#)]
6. Suri, S.; Vijay, R. An AES-chaos-based hybrid approach to encrypt multiple images, recent developments in intelligent computing, communication and devices. In *Advances in Intelligent Systems and Computing*; Springer: Singapore, 2017; Volume 555.
7. Hraoui, S.; Gmira, F.; Jarar, A.O.; Satori, K.; Saaidi, A. Benchmarking AES and chaos based logistic map for image encryption. In *Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications*, AICCSA, Ifrane, Morocco, 27–30 May 2013.
8. Arab, A.; Rostami, M.J. An image encryption method based on chaos system and AES algorithm. *J. Supercomput.* **2019**, 75, 6663–6682. [[CrossRef](#)]
9. Kotel, S.; Zeghid, M.; Tourki, R.; Machhout, M.; Baganne, A. High-Level Implementation of a Chaotic and AES Based Crypto-System. *J. Circuits Syst. Comput.* **2017**, 26, 1750122.
10. Yang, C.H. Symplectic Synchronization of Lorenz-Stenflo system with uncertain chaotic parameters via adaptive control. *Abstr. Appl. Anal.* **2013**, 2013. [[CrossRef](#)]
11. Abbas, A.M. Image encryption based on independent component Analysis and Arnold's cat map. *Egypt. Inform. J.* **2016**, 17, 139–146. [[CrossRef](#)]
12. Pecora, L.M.; Carroll, T.L. Synchronization in chaotic systems. *Phys. Rev. Lett.* **1990**, 64, 821–824. [[CrossRef](#)] [[PubMed](#)]
13. Njah, A.N. Tracking control and synchronization of the new hyperchaotic Liu system via backstepping techniques. *Nonlinear Dyn.* **2010**, 61, 1–9. [[CrossRef](#)]

14. Yu, Y.; Li, H.X. Adaptive hybrid projective synchronization of uncertain chaotic systems based on backstepping design. *Nonlinear Anal. Real World Appl.* **2011**, *12*, 388–393. [CrossRef]
15. Kuo, C.L. Design of a fuzzy sliding-mode synchronization controller for two different chaos systems. *Comput. Math. Appl.* **2011**, *61*, 2090–2095. [CrossRef]
16. Yau, H.T.; Kuo, C.L.; Yan, J.J. Fuzzy sliding mode control for a class of chaos synchronization with uncertainties. *Int. J. Nonlinear Sci. Numer. Simul.* **2006**, *7*, 333–338. [CrossRef]
17. Pai, M.C. Global synchronization of uncertain chaotic systems via discrete-time sliding mode control. *Appl. Math. Comput.* **2014**, *228*, 663–671. [CrossRef]
18. Heron, S. Advanced encryption standard (AES). *Netw. Secur.* **2009**, *12*, 8–12. [CrossRef]
19. Using AES Encryption in CC111xFx and CC251xFx. Application Report, SWRA172C-August 2008-Revised March 2015, Copyright © 2008–2015. Texas Instruments Incorporated. Available online: <http://www.ti.com/lit/er/er001> (accessed on 30 January 2021).
20. Ashokkumar, C.; Venkatesh, M.B.S.; Giri, R.P.; Roy, B.; Menezes, B. An error-tolerant approach for efficient AES key retrieval in the presence of cacheprefetching-experiments, results, analysis. *Sadhana* **2019**, *44*, 88. [CrossRef]
21. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510. [CrossRef] [PubMed]
22. Miller, D.A.; Grassi, G. A discrete generalized hyperchaotic Henon map circuit. In Proceedings of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems (MWSCAS), Dayton, OH, USA, 14–17 August 2001.
23. Liao, T.L.; Wan, P.Y.; Chien, P.C.; Liao, Y.C.; Wang, L.K.; Yan, J.J. Design of high-security USB flash drives based on chaos authentication. *Electronics* **2018**, *7*, 82. [CrossRef]
24. Gilbert, H.; Handschuh, H. *Security Analysis of SHA-256 and Sisters*, SAC 2003, LNCS 3006; Springer: Berlin/Heidelberg, Germany, 2004; pp. 175–193.
25. Delfs, H.; Knebl, H. *Symmetric-Key Encryption-Introduction to Cryptography: Principles and Applications*; Springer: Berlin/Heidelberg, Germany, 2007; ISBN 9783540492436.
26. Rukhin, A.; Soto, J.; Nechvatal, J.; Smid, M.; Barker, E. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*; Booz-Allen and Hamilton Inc.: McLean, VA, USA, 2001.
27. Kolassa, S.; Schütz, W. Advantages of the MAD/Mean Ratio over the MAPE. *Foresight: Int. J. Appl. Forecast.* **2007**, *6*, 40–43.
28. Pathria, R.K.; Beale, P.D. *Statistical Mechanics*, 3rd ed.; Academic Press: Cambridge, MA, USA, 2011; p. 51, ISBN 978-0123821881.