Multimedia Systems (2022) 28:1793–1808
https://doi.org/10.1007/s00530-022-00950-6

REGULAR ARTICLE

Novel design of cryptosystems for video/audio streaming via dynamic synchronized chaos-based random keys

Chih-Hsueh Lin[1] · Guo-Hsin Hu[1,2] · Jie-Sheng Chen[3] · Jun-Juh Yan[3] · Kuang-Hui Tang[3]

Received: 17 June 2021 / Accepted: 23 April 2022 / Published online: 21 May 2022

Abstract

In this paper, a novel chaos-based cryptosystem is proposed to ensure the communication security of video/audio streaming in the network environment. Firstly, by the proposed synchronization controller for the master and slave chaotic systems, respectively, embedded in the transmitter and receiver, the cryptosystem can generate the synchronized and dynamic chaotic random numbers at the transmitter and receiver simultaneously. Then integrating the chaotic random numbers with SHA3-256 (Secure hash algorithm 3), the design of synchronized dynamic key generators (SDKGs) is completed. Continuously, we can apply the SDKGs to encrypt/decrypt streaming audio/video data. In our design, we introduce the AES CFB (Advanced encryption standard cipher feedback) encryption algorithm with SDKGs to encrypt the video/audio streaming. Then the cipher-text is transmitted to the receiver via the network public channel and it can be fully decrypted with the dynamic random keys synchronously generated at the receiver. A duplex audio/video cryptosystem is realized to illustrate the performance and feasibility of this proposed research. Finally, many tests and comparisons are performed to stress the quality of random sequences generated by proposed SDKGs.

Keywords Chaos synchronization · Dynamic key generator · Secure communication · Video/audio streaming · AES CFB

1 Introduction

Due to the impact of COVID-19, to reduce the risk of infection caused by conversation contact, people are accustomed to using the wireless network for communication from chats to high-secret video conferences. Therefore, how to ensure the security of data exchange in the network environment is a very important issue. In traditional encryption algorithms, such as RSA, AES, ECC, etc., they all rely on the complexity of static keys and encryption algorithms to strengthen the confusion of cipher-text to protect private information [1]. In such a symmetric encryption method, a static key is used. However, when the static key is cracked, information will be fully exposed. Furthermore, due to the ultra-high computing speed of IBM's quantum computer with 53 qubits, the time required for brute-force attack is greatly reduced. This has become a major concern of secure communication in the near future [2]. On the other hand, a chaotic system can generate a large number of random signals in a very short period, and these random signals are dynamic and unpredictable. These chaos-based dynamic random keys, combined with traditional symmetric encryption, will make it difficult to crack. In addition, the chaos-based cryptosystem has not the problem of key storage, because the random keys are dynamically generated and updated by the chaotic systems. A chaotic system is a non-periodic dynamical system with random state responses. The state responses have the characteristics of nonlinear, unpredictable trajectory, sensitivity

Communicated by J. Dittmann.

✉ Jun-Juh Yan
jjyan@ncut.edu.tw

Chih-Hsueh Lin
cslin@nkust.edu.tw

Guo-Hsin Hu
guohsin@mail.mirdc.org.tw

[1] Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung 80778, Taiwan

[2] Department of Industrial Upgrading Service, Metal Industries Research and Development Centre, Kaohsiung 81160, Taiwan

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

to initial values, butterfly effect and strange attractors. The concept of hyper-chaotic systems was first proposed by Rössler in 1979 [3]. Hyper-chaotic systems have multiple strange attractors with positive Lapunov exponents. The unpredictable property is useful for secure communication and the application of communication security has attracted many researchers to invest in the literature [4, 5]. Among the reports [4, 5] on the encryption application, the design of a random number generator is very important. Generally speaking, random number generators have two types, namely true random number generators (TRNGs) and pseudo-random number generators (PRNGs) [6, 7]. True random numbers are unpredictable signals that usually exist in nature, such as electromagnetic noise and thermal noise. However, to extract true random numbers from the natural environment, additional sensing circuits must be used to obtain and convert signals, which is inconvenient and high in cost. Therefore, to reduce the cost, different artificial methods are proposed to generate pseudo-random numbers with the random property as the TRN as possible [8–10]. However, due to the difficulty of modeling, the real random number is very difficult to achieve synchronization control and apply to communication security. On the contrary, with the deterministic mathematical model of chaotic systems, it becomes possible to apply the chaos-based random numbers to the design of cryptosystems through synchronization controller design. In 1990, OGY (Ott, Grebogi and Yorke) [11] proposed the related research on controlling chaos. Then, Pecora and Carroll [12] tried to synchronize two chaotic systems with different initial values to have the same random state responses. Since then, there have been many studies focusing on the synchronization of chaotic systems [13, 14] and various control methods had been introduced for synchronization, such as sliding mode control [15, 16], optimal control [17], adaptive control [18], discrete sliding mode control [19] and fuzzy control [20], etc. The synchronization control can be utilized to generate the synchronized dynamic keys and then further design a high secure cryptosystem to guarantee communication security.

Motivated by the aforesaid, this study aims to develop a novel chaos-based cryptosystem to ensure the communication security of video/audio streaming in the network environment. To obtain high-quality random numbers, we use the generalized 4-dimensional Lorenz-Stenflo (4D LS) hyper-chaotic system to design the random number generator. The generalized 4D LS hyper-chaotic system is an advanced system based on the 3-dimensional continuous Lorenz system, which can dynamically generate four unpredictable random numbers [21, 22]. We firstly discretize the 4D LS hyper-chaotic system as a discrete model [23] such that it becomes easy to implement with microcontroller, and then design a discrete synchronization controller such that we can synchronize the master and slave chaotic systems to obtain the same random numbers simultaneously at both transmitter and receiver. It is worth mentioning that, in our design, the chaotic states and control signals are not necessary to fully expose in the public channel, so the security of the system can be promoted. After obtaining the synchronized chaotic random numbers, to promote the randomness quality of random numbers, we further integrate with the SHA3-256 algorithm [24] to complete the design of synchronized dynamic key generators (SDKGs) with a fixed length of 256 bits, which can be used for communication encryption. With the proposed SDKGs, we can complete the design of the dynamic cryptosystem for video/audio streaming. In this dynamic cryptosystem, audio/video signals are captured from the local microphone and camera, and then the proposed SDKGs are used to replace the static key of the traditional symmetric encryption algorithm AES CFB [25], to dynamically encrypt the data, and then send the ciphertext through the public channel. When the receiver receives the ciphertext, it will use the synchronized dynamic random keys of SDKGs with the AES CFB to decrypt the ciphertext, and finally recover the original audio/video data. The statistical analysis, histogram, connected component analysis [26, 27], information entropy [28], keyspace analysis and correlation indexes were calculated and analyzed through simulation experiments and comparisons to highlight the capability and feasibility of this design method.

This paper is organized as follows. Section 2 formulates chaos synchronization of 4D LS systems and the SDKGs implementation in the network environment. The chaos-based cryptosystem with SDKGs is realized in Sect. 3. Section 4 evaluates the security of the proposed chaos-based random key generator and chaos-based cryptosystem. Finally, we give brief conclusions in Sect. 5.

2 Synchronization of master–slave 4D LS hyper-chaotic systems and design of SDKGs

In this research, the design of a dynamic random key generator and its synchronization control are important core technologies. Therefore, we first discuss the synchronization controller design and implementation of the master–slave chaotic systems. Simultaneously, to facilitate the low-cost realization with digital microcontrollers, discrete chaotic systems will be considered. In the following, we will introduce the 4D LS hyper-chaotic system to discuss, of course, the technology developed in this paper can be extended and applied to different chaotic systems. The state equation of the generalized 4D LS system can be described as follows:

$$\begin{aligned}\dot{x}_1(t) &= -ax_1(t) + ax_2(t) + \lambda x_3(t)\\ \dot{x}_2(t) &= dx_1(t) + \gamma x_2(t) - x_1(t)x_4(t),\\ \dot{x}_3(t) &= -cx_1(t) - x_3(t)\\ \dot{x}_4(t) &= x_1(t)x_2(t) - bx_4(t)\end{aligned} \tag{1}$$

where a, b, c, d, γ and λ are parameters of the system (1). x_1, x_2, x_3 and x_4 are the state variables. To facilitate the realization with the digital devices or components, we discretize the continuous system to obtain a corresponding discrete system. The following describes the method of system discretization. For a continuous-time chaotic system described by

$$\dot{x}(t) = Ax(t) + Bg(x(t), t), \tag{2}$$

where $x(t) \in R^n$ is the state vector;$g(x(t), t) \in R^r$ is the nonlinear function of systems. Matrices A and B are controllable. Then the discrete system corresponding to the system (2) can be obtained by

$$x_d(k+1)T = Gx_d(kT) + Hg\left(x_d(kT)\right), \tag{3}$$

where $G = e^{AT}$ and $H = \left[G - I_n\right]A^{-1}B$[23], T is the sampling time. Obviously the 4D LS hyperchaotic system (1) can be rearranged as

$$\dot{x}(t) = \begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\\ \dot{x}_3(t)\\ \dot{x}_4(t)\end{bmatrix} = Ax(t) + B\begin{bmatrix}-x_1(t) & x_4(t)\\ x_1(t) & x_2(t)\end{bmatrix}, \tag{4}$$

where $x(t) \in R^4$ and $x(t) = \left[x_1(t)\ x_2(t)\ x_3(t)\ x_4(t)\right]^T$, $A \in R^{4\times 4}$ and $B \in R^{4\times 2}$ are obtained as

$$A = \begin{bmatrix}-a & a & \lambda & 0\\ d & \gamma & 0 & 0\\ -c & 0 & -1 & 0\\ 0 & 0 & 0 & -b\end{bmatrix}, B = \begin{bmatrix}0 & 0\\ 1 & 0\\ 0 & 0\\ 0 & 1\end{bmatrix}, \tag{5}$$

According to [23], the continuous-time 4D LS hyperchaotic system (1) can be discretized as:

$$x_d(k+1) = Gx_d(k) + H\begin{bmatrix}-x_{d1}(k) & x_{d4}(k)\\ x_{d1}(k) & x_{d2}(k)\end{bmatrix}, \tag{6}$$

where $G \in R^{4\times 4}$ and $H \in R^{4\times 2}$ can be calculated by $G = e^{AT}$ and $H = \left[G - I_n\right]A^{-1}B$. $x_d(k) = \left[x_{d1}(k)\ x_{d2}(k)\ x_{d3}(k)\ x_{d4}(k)\right]^T$ is the state vector of the discrete system (6).

Now we utilize the discretized 4D LS system (6) to design the 4D LS RNGs which can be synchronized by the synchronization controller. First, master and slave 4D LS systems are given, respectively, as (6) and (7).

$$y_d(k+1) = Gy_d(k) + H\begin{bmatrix}-y_{d1}(k)y_{d4}(k)\\ y_{d1}(k)y_{d2}(k)\end{bmatrix} + H\begin{bmatrix}u_1(k)\\ u_2(k)\end{bmatrix}, \tag{7}$$

where $x_d(k) = \left[x_{d1}(k)\ x_{d2}(k)\ x_{d3}(k)\ x_{d4}(k)\right]^T$ is the state vector of master system (7) embedded in the transmitter; $y_d(k) = \left[y_{d1}(k)\ y_{d2}(k)\ y_{d3}(k)\ y_{d4}(k)\right]^T$ is the state vector of slave system (8) embedded in the receiver; $u_1(k)$ and $u_2(k)$ are the control inputs designed later to ensure synchronization (i.e. $x_d(k) = y_d(k)$) between master and slave systems. By using matrices in (5) with parameters $a = 11.0, b = 2.9, c = 5, d = 23, \gamma = -1, \lambda = 1.9$, sampling time$T = 0.001$, and the formulas of $G = \mathrm{e}^{AT}$ and $H = \left[G - I_n\right]A^{-1}B$ [23], matrices G and H can be calculated as

$$G = \begin{bmatrix}0.9892 & 0.0109 & 0.0019 & 0.0000\\ 0.0229 & 0.9991 & 0.0000 & 0.0000\\ -0.0050 & 0.0000 & 0.9990 & 0.0000\\ 0.0000 & 0.0000 & 0.0000 & 0.9971\end{bmatrix}, H = \begin{bmatrix}0.0000 & 0.0000\\ 0.0010 & 0.0000\\ 0.0000 & 0.0000\\ 0.0000 & 0.0010\end{bmatrix} \tag{8}$$

Obviously, the matrix pair (G, H) is controllable. Then we define the error state as

$$e_{di} = y_{di} - x_{di}, \tag{9}$$

where $i = 1, 2, 3, 4$, Then from (6), (7), (9), we have the error dynamics as

$$\begin{aligned}e_d(k+1) &= Ge_d(k)\\ &+H\left(\begin{bmatrix}x_{d1}(k)x_{d4}(k) - y_{d1}(k)y_{d4}(k)\\ -x_{d1}(k)x_{d2}(k) + y_{d1}(k)y_{d2}(k)\end{bmatrix} + \begin{bmatrix}u_1(k)\\ u_2(k)\end{bmatrix}\right),\end{aligned} \tag{10}$$

where $e_d(k) = \left[e_{d1}(k)\ e_{d2}(k)\ e_{d3}(k)\ e_{d4}(k)\right]^T$。

The control input in the slave system (8) is designed as

$$\begin{bmatrix}u_1(k)\\ u_2(k)\end{bmatrix} = u_c(k) - \begin{bmatrix}x_{d1}(k)x_{d4}(k) - y_{d1}(k)y_{d4}(k)\\ -x_{d1}(k)x_{d2}(k) + y_{d1}(k)y_{d2}(k)\end{bmatrix}, \tag{11}$$

where $u_c(k) = -Ke_d(k)$, $K \in R^{2\times 4}$ is the designed gain matrix.

Substituting (11) into (10), we have

$$e_d(k+1) = (G - HK)e_d(k). \tag{12}$$

Since (G, H) is controllable, we can arbitrarily assign the eigenvalues of the matrix $(G - HK)$. Thus, we can apply the pole assignment technology to design the matrix K such that $\left|\mathrm{Re}\lambda_i(G - HK)\right| < 1, i = 1, 2, 3, 4$ and the error dynamics (12) is asymptotically stable. Obviously, from (12), the convergence speed of $e_d(k)$ can be assigned and predicted with the eigenvalues of matrix $(G - HK)$.

From the above discussion, it reveals that the controller (11) can asymptotically synchronize the master and slave chaotic systems. However, in the realization of the controller, to avoid the control information being exposed in the network environment, we disassemble the controller (11)

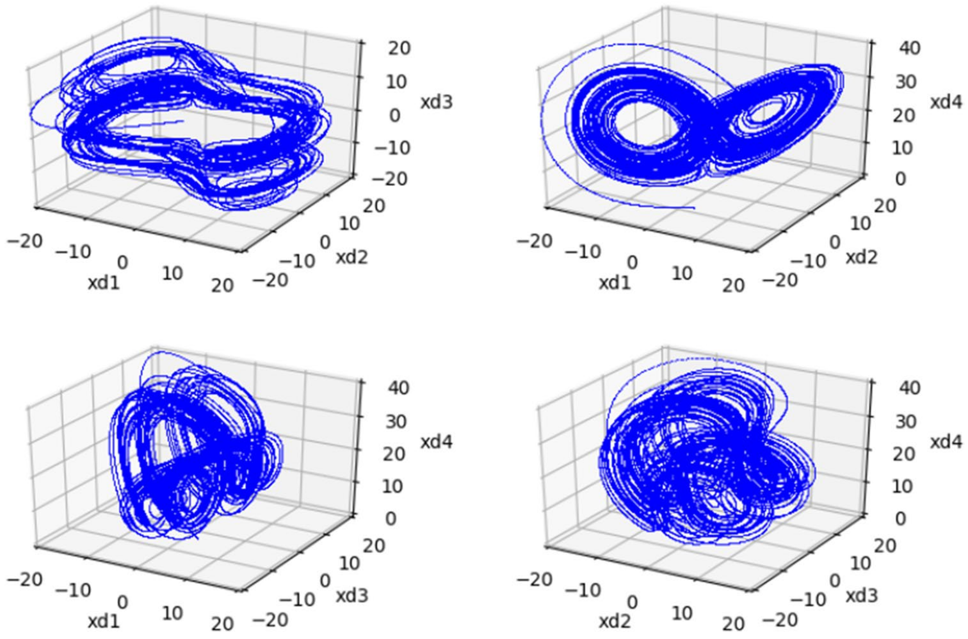


Fig. 1 Strange attractor of the discrete 4D LS hyper-chaotic system

into two parts, which are calculated separately at the transmitter and receiver, and then the complete control input signal is assembled at the receiver to achieve synchronization. The design is given as follows

$$\begin{bmatrix} u_1(k)\\ u_2(k)\end{bmatrix} = \overbrace{Kx_d(k) + \begin{bmatrix} -x_{d1}(k) & x_{d4}(k) \\ x_{d1}(k) & x_{d2}(k)\end{bmatrix}}^{u_m(k)} - \overbrace{Ky_d(k) + \begin{bmatrix} y_{d1}(k) & y_{d4}(k)\\ -y_{d1}(k) & y_{d2}(k)\end{bmatrix}}^{u_s(k)}, \tag{13}$$

where $u_m(k) = \begin{bmatrix} u_{m1} & u_{m2} \end{bmatrix}^T$ and $u_s(k) = \begin{bmatrix} u_{s1} & u_{s2} \end{bmatrix}^T$ are calculated, respectively, at the transmitter and receiver. In the following, we utilize the simulation tool of MATLAB to verify the control design discussed above. In this simulation, we first assign the eigenvalues of $\lambda(G - HK) = \begin{bmatrix} 0.1 & -0.1 & 0.08 & 0.09 \end{bmatrix}^T$ and the corresponding gain matrix K can be easily obtained using the pole assignment method with the command of 'Place' in the Matlab toolbox.

$$K = \begin{bmatrix} 0.0216 & 0.0003 & -1.6817 & 0 \\ 0 & 0 & 0 & 0.0001 \end{bmatrix} \times 10^7 \tag{14}$$

In this simulation, the initial conditions were selected as $x_{d1} = -1.0$, $x_{d2} = -3.0$, $x_{d3} = 2.0$, $x_{d4} = -5.0$, $y_{d1} = -1.1$, $y_{d2} = -3.0$, $y_{d3} = 2.0$ and $y_{d4} = -4.0$. Figures 1, 2, 3, 4, 5 show the simulation results. Figure 1 shows the strange attractor of the discrete version corresponding to the continuous 4D LS hyper-chaotic system in the transmitter. It confirms that the continuous hyper-chaotic system can be simulated by the corresponding discrete version. Figure 2 shows the state responses of the slave system in the receiver. Figure 2 also shows that the state of the hyper-chaotic system is random and cannot be predicted. Figure 3 shows the synchronization error between the master and slave systems. Figures 4 and 5 show, respectively, the control inputs and the state responses of master–slave hyper-chaotic systems. We can see that the synchronization errors converge to zero due to the control input. It means that we can obtain the same chaotic random number simultaneously at both transmitter and receiver.

After verifying the synchronization, the design of chaos-based SDKGs is shown in Fig. 6.

In Fig. 6, $u_{\mathrm{m}1}(k)$ and $u_{\mathrm{m}2}(k)$ are firstly calculated at the transmitter (Master systems) and transmitted to the receiver through the public channel. At the same time, $u_{s1}(k)$ and $u_{s2}(k)$ are also obtained at the receiver (Slave systems). And then the receiver integrates the received $u_{\mathrm{m}1}(k)$ and $u_{\mathrm{m}2}(k)$ with the local signals $u_{s1}(k)$ and $u_{s2}(k)$ to obtain the synchronization controller in (14) and achieve synchronization of master and slave chaotic systems embedded in the transmitter and receiver, respectively. Finally, the synchronized chaotic random numbers are inputted to the SHA3-256 algorithm to generate synchronized random number sequence with a fixed length of 256 bits, respectively, at the transmitter and receiver. Such SDKGs will be applied for data encryption applications.

3 Cryptosystems for video/audio streaming via SDKGs

In the previous discussion, due to the synchronization controller, the master and slave hyper-chaotic systems at the transmitter and receiver can be synchronized. Then,

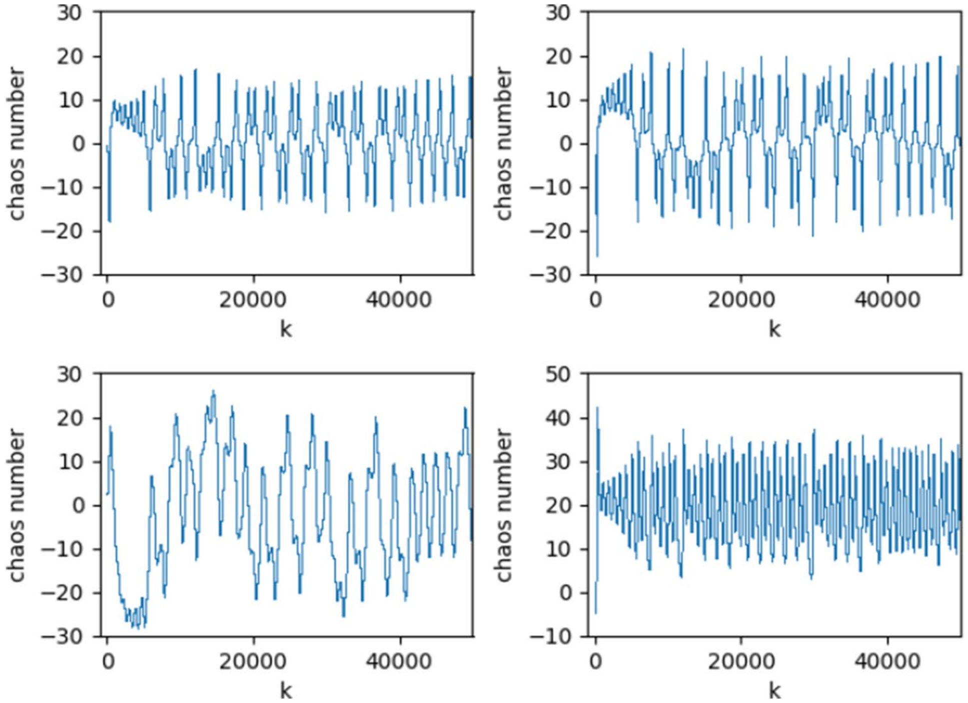


Fig. 2 State responses of the slave discrete 4D LS hyperchaotic system

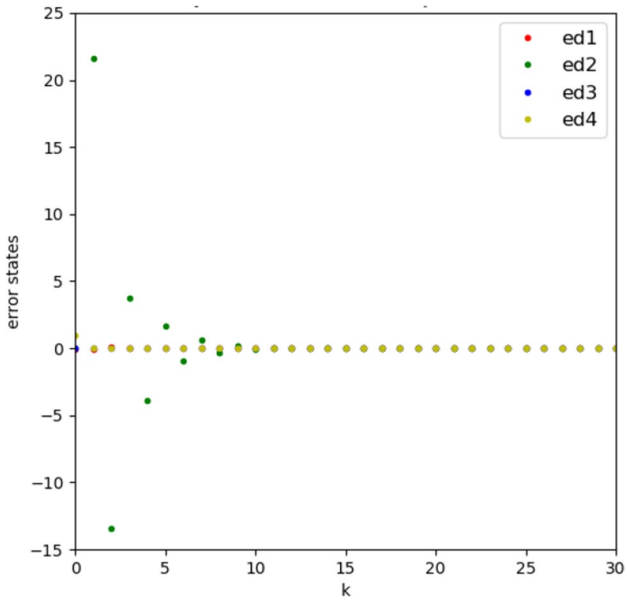


Fig. 3 The time responses of synchronization errors

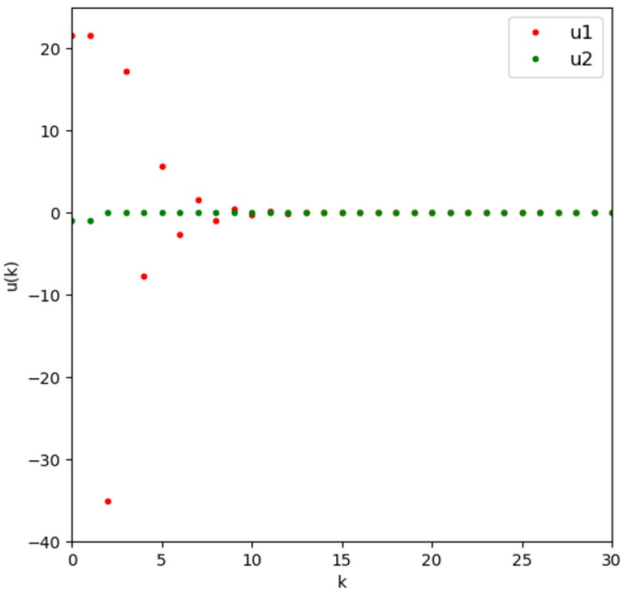


Fig. 4 The time responses of control inputs

through the avalanche effect of the SHA3-256 algorithm, a SDKG can be established. After obtaining the synchronized dynamic keys, we can apply them to the communication security of video/audio streaming data. The structure of the cryptosystem is shown in Fig. 7, which includes Lorenz-Stenflo SDKGs and encryption/decryption mechanisms. After the hyper-chaotic system of User A is synchronized with that of user B, both users can send the data of captured images and sounds to the encryption/decryption mechanisms. The modified AES CFB algorithm with synchronized dynamic keys is used for encryption /decryption calculations, and then the ciphertext is sent to the receiver through the network for decryption and finally the original plaintexts of streaming video/audio are obtained at the receiver. The images are pre-converted to JPEG format to reduce the load of network transmission.

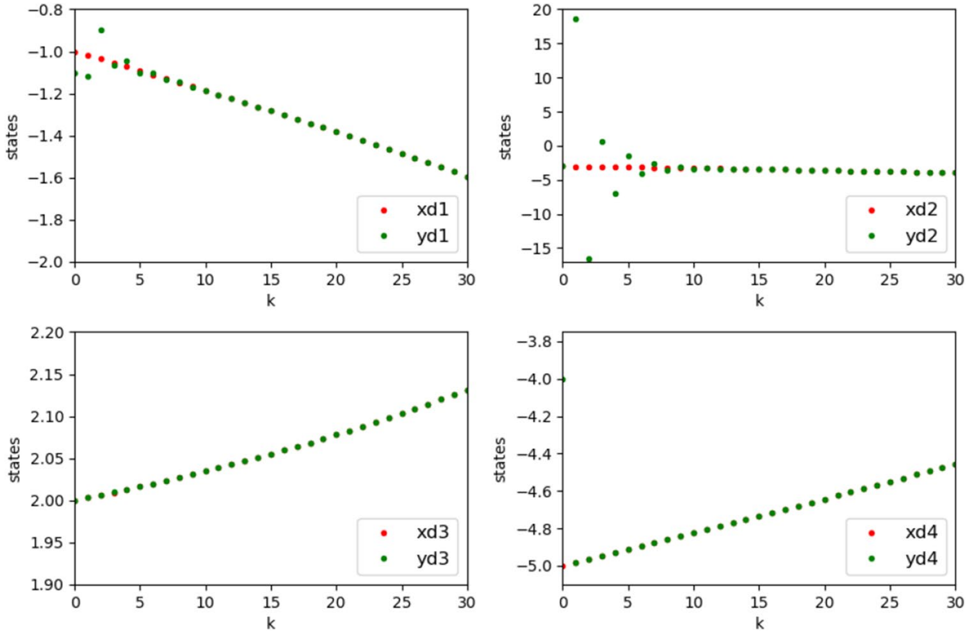


Fig. 5 State responses of the master–slave discrete 4D LS systems

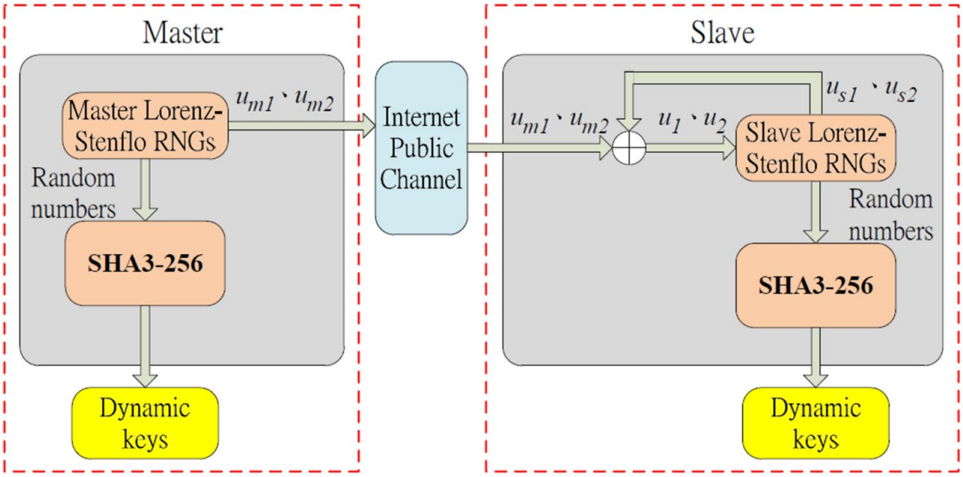


Fig. 6 The structure of SDKGs

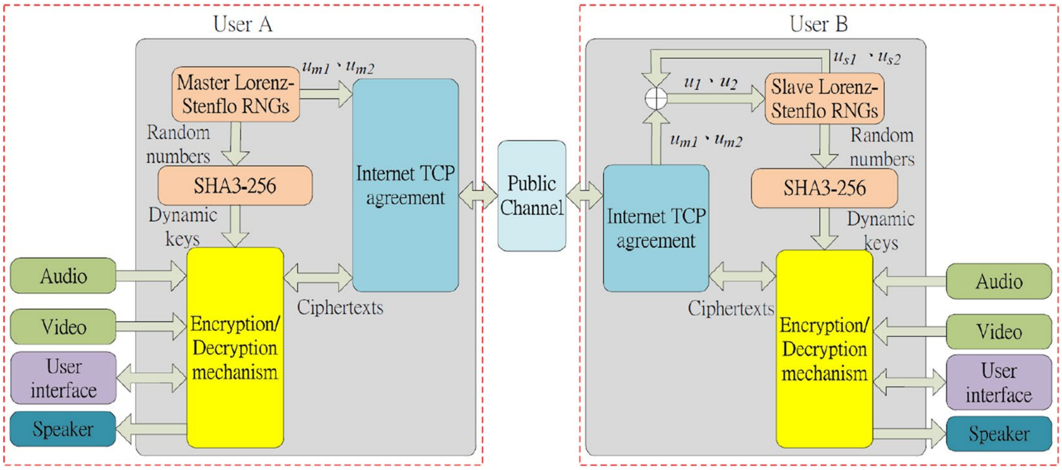


Fig. 7 The structure of cryptosystem

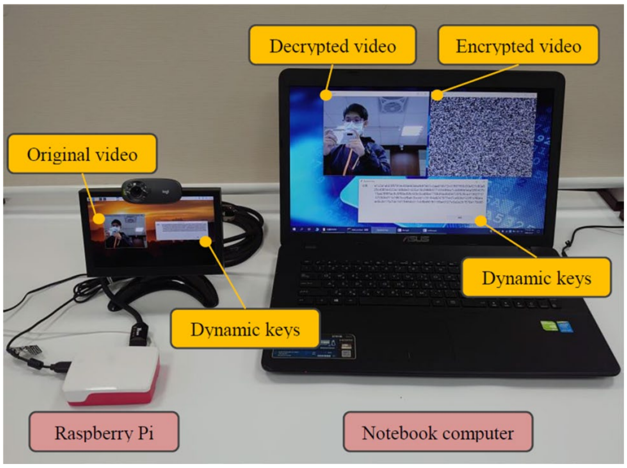
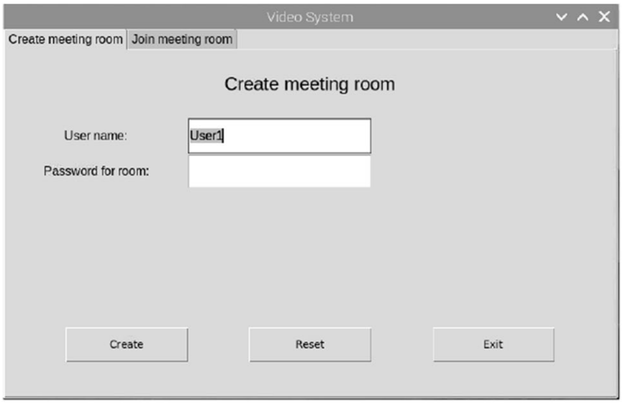


Fig. 8 The realization of the cryptosystem for video/audio streaming

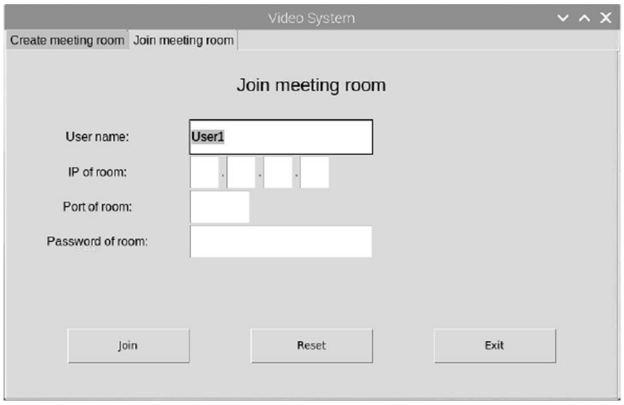


Create meeting room

User name: User1

Password for room:

Fig. 9 Creating a meeting room



Join meeting room

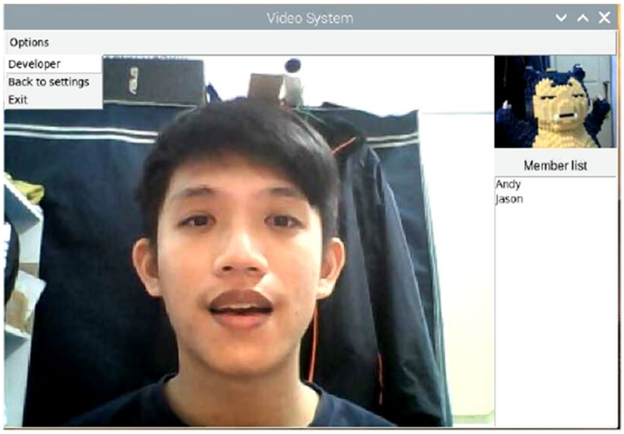
User name: User1

IP of room: . . .

Port of room:

Password of room:

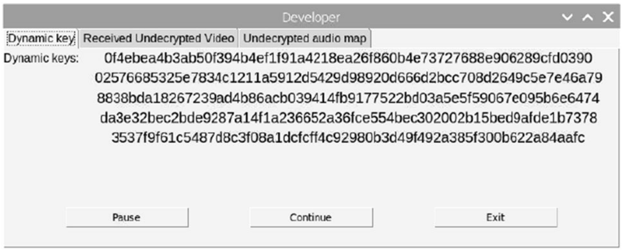
Fig. 10 Joining a meeting room



Member list

Andy
Jason

Fig. 11 Window of the meeting room



```
Dynamic keys:   0f4ebea4b3ab50f394b4ef1f91a4218ea26f860b4e73727688e906289cfd0390
               02576685325e7834c1211a5912d5429d98920d666d2bcc708d2649c5e7e46a79
                8838bda18267239ad4b86acb039414fb9177522bd03a5e5f59067e095b6e6474
                da3e32bec2bde9287a14f1a236652a36fce554bec302002b15bed9afde1b7378
                 3537f9f61c5487d8c3f08a1dcfcff4c92980b3d49f492a385f300b622a84aafc
```

Fig. 12 Random dynamic keys

The realization of the system is shown in Fig. 8. We use the Raspberry Pi and the computer as the platforms for User A and User B, respectively. Then use the network camera to capture audio and video, as shown in Fig. 7, and transmit the real-time video and audio of both users through the TCP (Transmission Control Protocol) protocol with the Internet channel. The data is encrypted/decrypted through the synchronous dynamic keys and the AES CFB algorithm designed in this paper to achieve real-time video/audio data secure transmission.

In this cryptosystem as shown in Fig. 8, we further design a user interface. Users can build or join the streaming video conference room with their name and password as shown in Figs. 9 and 10, respectively. After confirming, users enter the window of the meeting room as shown in Fig. 11. The left image is the other user's image, and the upper right image is the user’s image. The bottom right is a list of all the users in this meeting room. Users can click ‘Leave’ in the upper left option to leave this meeting room. In addition, we also design the developer window to observe the system information. As shown in Figs. 12, 13, 14, 15, the dynamic keys, encrypted video, original audio signal and encrypted audio signal are displayed, respectively. The latest five dynamic random keys generated by

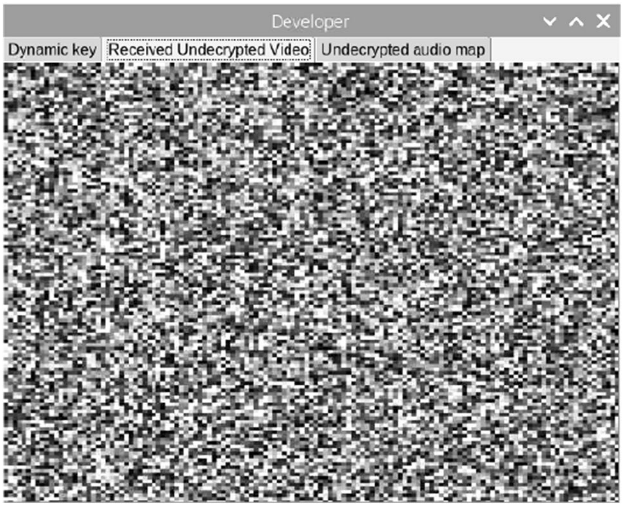


Fig. 13 Encrypted image

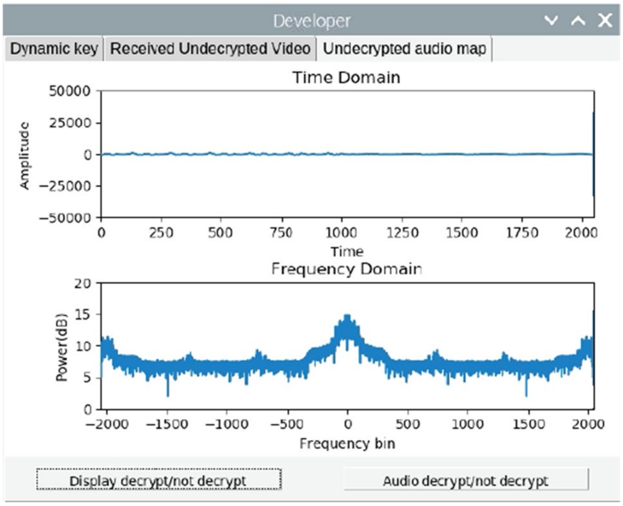


Fig. 14 The original audio signal

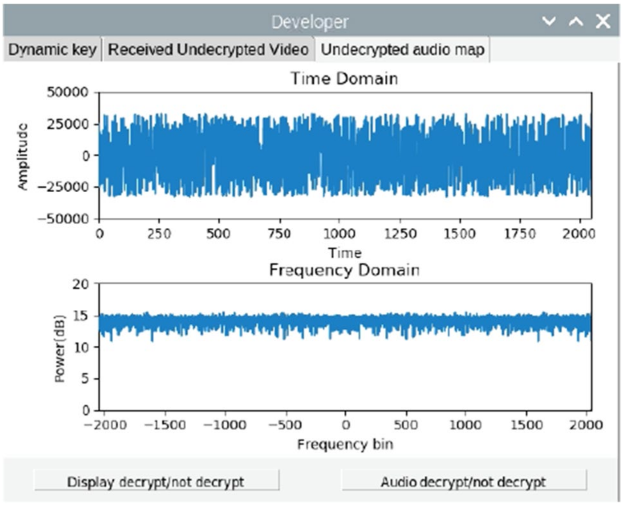


Fig. 15 The encrypted audio signal

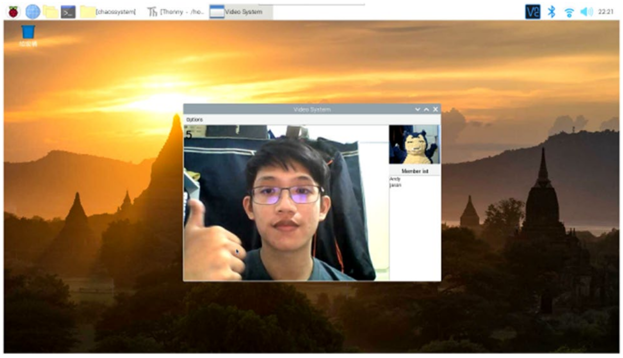


Fig. 16 Window of user A

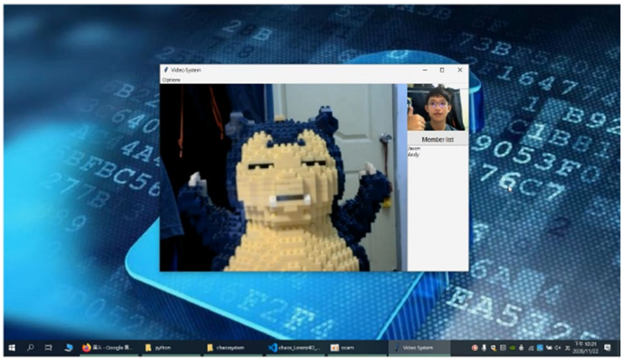


Fig. 17 Window of user B

SDKGs are shown in Fig. 12, and we can click ‘Pause’ or ‘Continue’ to update the dynamic key display; Fig. 13 shows the received encrypted image on the receiver, and it can be found that we cannot extract any information about original video image; the upper figure in Fig. 14 shows the received original audio data and the lower figure shows the Fourier spectrum of the audio data; the upper in Fig. 15 shows the un-decrypted audio signal, and the lower shows the Fourier spectrum of the un-decrypted sound signal, it can be observed that if it has not been decrypted, the sound is chaotic, and the Fourier spectrum is broad. Finally, in Figs. 16 and 17, we can have the actual video windows of the encryption/decryption in this system. The above results demonstrate that the proposed system with a synchronized 4D LS dynamic key generator can successfully perform dynamic encryption/decryption for video/audio stream data.

4 Security analysis of SDKGs and chaos-based cryptosystem

This article integrates synchronized master–slave hyperchaotic systems with the SHA3-256 algorithm to generate dynamic random keys. Then the dynamic keys are combined with the AES CFB encryption algorithm to complete the cryptosystem for the real-time video/audio stream. To ensure the security of this encryption technology, the statistical analysis, histogram, connected component analysis(CCA), information entropy (IE), and key spaces are calculated in the following. To test the randomness quality of the encrypted streaming video, a 512×512 picture is prepared as shown in Fig. 18a. In actual application, to reduce the network transmission, the image will be converted from the pixel array to the JPEG format. After the format conversion, the file size is reduced from 786,432 bytes to 54,236 bytes. It can reduce the processing time for encryption/decryption. Then the JPEG file is encrypted by AES CFB with dynamic keys. The results are shown in Fig. 18b, c with one static key and five dynamic keys, respectively. It can be found that the encrypted images have no obvious features, and finally, it is decrypted and converted back to a pixel array as shown in Fig. 18d. Furthermore, to test the security of the actual video streaming, we continuously capture 12 pictures as shown in Fig. 19 from the video streaming. Figures 19a and b are the first and last captured pictures, respectively. All pictures are with a size of 819,234 Bytes. Also, a pre-recorded audio file with the size of 819,234 Bytes is used to test the randomness.

4.1 NIST analysis

In this section, we introduce the National Institute of Standards and Technology (NIST) test suite [8] to evaluate the randomness of the dynamic random keys and encrypted data. This evaluation standard contains 15 items and the test result for every test item is called the p value. When the p value > 0.01, it means that the test item has passed. With a larger p value, it means the better randomness of the test data. We select the dynamic keys generated with the state x_{d1} of the chaotic system. The test results in Table 1 were obtained with a stream length of 6,553,872 bits and a bitstream count of 30. The captured pictures in Fig. 19 and pre-recorded sound file are encrypted with AES CFB and five dynamic keys. Obviously, all the test results shown in Table 1 pass the tests. The performance results of the traditional AES CFB only using a static key are shown in Table 2. According to the results in Tables 1

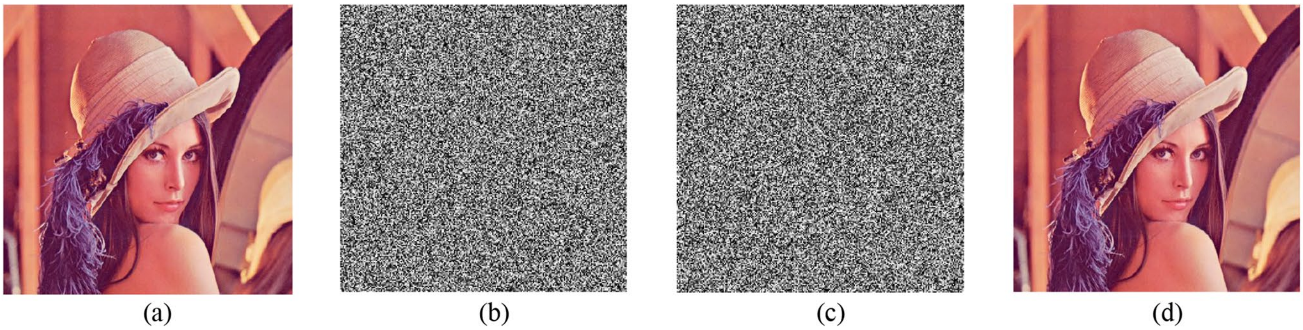


Fig. 18 Picture files (**a**) original image (**b**) encrypted image by one key (**c**) encrypted image by five keys (**d**) decrypted image

Fig. 19 Real-time 12 images captured (**a**) The first one captured; **b** The last one captured

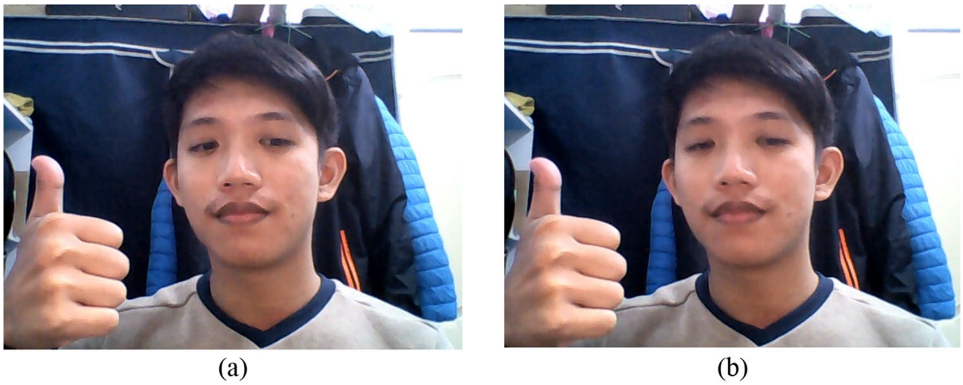


Table 1 NIST SP 800-22 test results of dynamic random keys and pre-recorded file

Statistical tests	p value ($N=6553872$bits)					
	Lorenz-Stenflo dynamic random keys x_{d1}-SHA3		Real picture encryption by five dynamic keys		Audio encryption by five dynamic keys	
Frequency	0.757041168321815	PASS	0.6110444100161085	PASS	0.6589255483728064	PASS
Block frequency	0.7383177081052352	PASS	0.92991215444075	PASS	0.7335514974228903	PASS
Runs	0.9066837727254307	PASS	0.8263184775182874	PASS	0.9545818671524448	PASS
Longest run	0.6163610211186337	PASS	0.4678881971400057	PASS	0.0581213714768202	PASS
Rank	0.3945721752108433	PASS	0.7715368176931643	PASS	0.0963522177689812	PASS
FFT	0.6062331649869583	PASS	0.8825927007306478	PASS	0.8600113743965508	PASS
Non overlapping template	0.9565908225403975	PASS	0.9999308432701147	PASS	0.9986014615967732	PASS
Overlapping template	0.9487989226049403	PASS	0.9304215198760762	PASS	0.3240886750671797	PASS
Universal	0.7722149324488802	PASS	0.6436355479448661	PASS	0.7049263631799445	PASS
Linear complexity	0.1696575433616641	PASS	0.3781427543370102	PASS	0.776556111067934	PASS
Serial	0.6910811818465875	PASS	0.1568080639957803	PASS	0.7143062057384221	PASS
Approximate entropy	0.9235856474055549	PASS	0.4145752284146443	PASS	0.9678361050026599	PASS
Cumulative sums	0.8947265993023332	PASS	0.8230982564174816	PASS	0.7978793460461064	PASS
Random excursions	0.344126023705281 5	PASS	0.1946342832910313	PASS	0.1125067188907348	PASS
Random excursions variant	0.2007686602112634	PASS	0.1002631331664804	PASS	0.1546203954515068	PASS
Total	9.920759343895819		9.130802388252448		8.912865258631756	

Table 2 NIST SP 800-22 test results of encrypted images and audio file with a static key

Statistical tests	p value ($N=6553872$bits)			
	Real picture encryption by one key (static key)		Audio encryption by one key (static key)	
Frequency	0.3291829148792813	PASS	0.4627295044790676	PASS
Block frequency	0.179264082869327	PASS	0.0963946278914413	PASS
Runs	0.0887672235857227	PASS	0.8400973273103713	PASS
Longest run	0.0620643443370354	PASS	0.2385709527832993	PASS
Rank	0.0283940348765471	PASS	0.5184402120053612	PASS
FFT	0.0000000000000000	FAIL	0.1129466953768003	PASS
Non overlapping template	0.9986483686566213	PASS	0.999980532401379	PASS
Overlapping template	0.7500930774379598	PASS	0.0657528921760780	PASS
Universal	0.8847786315112955	PASS	0.9055389220828538	PASS
Linear complexity	0.1760571799544020	PASS	0.3333733165763285	PASS
Serial	0.3040378078712666	PASS	0.7774485821850803	PASS
Approximate entropy	0.3114147932459213	PASS	0.7854697563072572	PASS
Cumulative sums	0.2869884170400594	PASS	0.6725861672529172	PASS
Random excursions	0.1276420264525448	PASS	0.1719450981187654	PASS
Random excursions variant	0.2624940429188451	PASS	0.1239030211557741	PASS
Total	4.789826945636829		7.105177608103776	

and 2, it shows that under the same conditions, the test results of the sound file and images using the dynamic keys are about 1.25 and 1.91 times, respectively, better than those with the static key. Furthermore, in Table 3, we compare the test results in this paper with those of other random number generators proposed in [29, 30]. The average score is as high as 7.31865, which is much higher than those obtained in [29, 30].

4.2 Histogram analysis

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 could not analyze statistical features of the original image from its cipher image. Each pixel in the picture is between 0 and 255. Use the image as shown in Fig. 18a for

Table 3 NIST SP 800–22 test results in [29, 30]

Statistical tests	*p* value (N=10^6 bits)								
	Lorenz-Stenflo RNG				RNG [29]	RNG [30]			
	x_{d1}-SHA3	x_{d2}-SHA3	x_{d3}-SHA3	x_{d4}-SHA3	z-4bit	x-8bit	y-8bit	z-8bit	u-8bit
Frequency	0.4942	0.4324	0.8299	0.1247	0.5445	0.2145	0.4879	0.8872	0.7455
Block frequency	0.6350	0.3786	0.9680	0.5251	0.4404	0.4978	0.3797	0.6798	0.8452
Runs	0.6979	0.3707	0.8452	0.9785	0.0514	0.2978	0.4879	0.0987	0.1879
Longest run	0.2535	0.4653	0.1544	0.1133	0.1896	0.7945	0.6617	0.3798	0.5579
Rank	0.3366	0.8265	0.5219	0.4531	0.3705	0.8798	0.4975	0.2648	0.3279
FFT	0.0304	0.4128	0.1481	0.9614	0.2438	0.0456	0.4521	0.1278	0.6159
Non overlapping template	1.0000	1.0000	0.9994	1.0000	0.0742	0.9785	0.0174	0.1245	0.0345
Overlapping template	0.8237	0.8135	0.6496	0.2735	0.6543	0.2898	0.6428	0.2895	0.1895
Universal	0.4491	0.8956	0.6401	0.5090	0.4279	0.8027	0.3788	0.4625	0.2985
Linear complexity	0.5089	0.8208	0.6956	0.6207	0.7117	0.2780	0.3470	0.6782	0.3722
Serial	0.0378	0.3290	0.7266	0.3382	0.3697	0.8796	0.2174	0.2879	0.7954
Approximate entropy	0.0818	0.3298	0.7271	0.3356	0.7380	0.5312	0.3789	0.4025	0.1820
Cumulative sums	0.2687	0.4034	0.5157	0.1389	0.8358	0.3798	0.2987	0.8749	0.6745
Random excursions	0.4406	0.0297	0.0225	0.1511	0.2700	0.3048	0.2789	0.1789	0.7924
Random excursions variant	0.2369	0.0487	0.2729	0.1826	0.5700	0.4278	0.3789	0.2879	0.1952
Total	6.2951	7.5568	8.7170	6.7057	6.4918	7.6022	5.9056	6.0249	6.8145
Total average	7.31865				6.4918	6.5868			

the test. First, the histogram analysis for the grayscale image of the original image is shown in Fig. 20a. Figure 20b shows the histogram analysis for the image in JPEG format. It is observed that both Figs. 20a and b have obvious differences in some specific pixel values. The histogram of the cipher image by one static key and five dynamics keys are shown in Figs. 20c and d. From Figs. 20c and d, it demonstrates that the encryption could effectively eliminate the differences between the histogram information.

To evaluate the uniform distribution of histograms, we calculate the variance of histogram [31] defined by

$$V(X) = \frac{1}{256^2} \sum_{i=0}^{255} \sum_{j=0}^{255} \frac{1}{2}\left(x_i - x_j\right)^2, \tag{15}$$

where$X = \left\{x_0, x_1, \cdots, x_{255}\right\}$, and x_i, x_j are the number of pixel values i, j, respectively.

Table 4 shows the variance of histograms in (15). The variance of the original image in Fig. 20a is extremely large, and it is reduced to 17,383 after being converted into JPEG format. When the traditional fixed key (static key) is used for the AES CFB encryption algorithm, it can be found that the variance is reduced to 232. However, with our encryption approach, the variance has been reduced to 202. The comparisons show that the proposed encryption method is effective to reduce the deviation of the picture and make it difficult to identify any features of the picture. Under the same conditions, compared with other proposed papers [32], we can find that our proposed encryption effect is very effective.

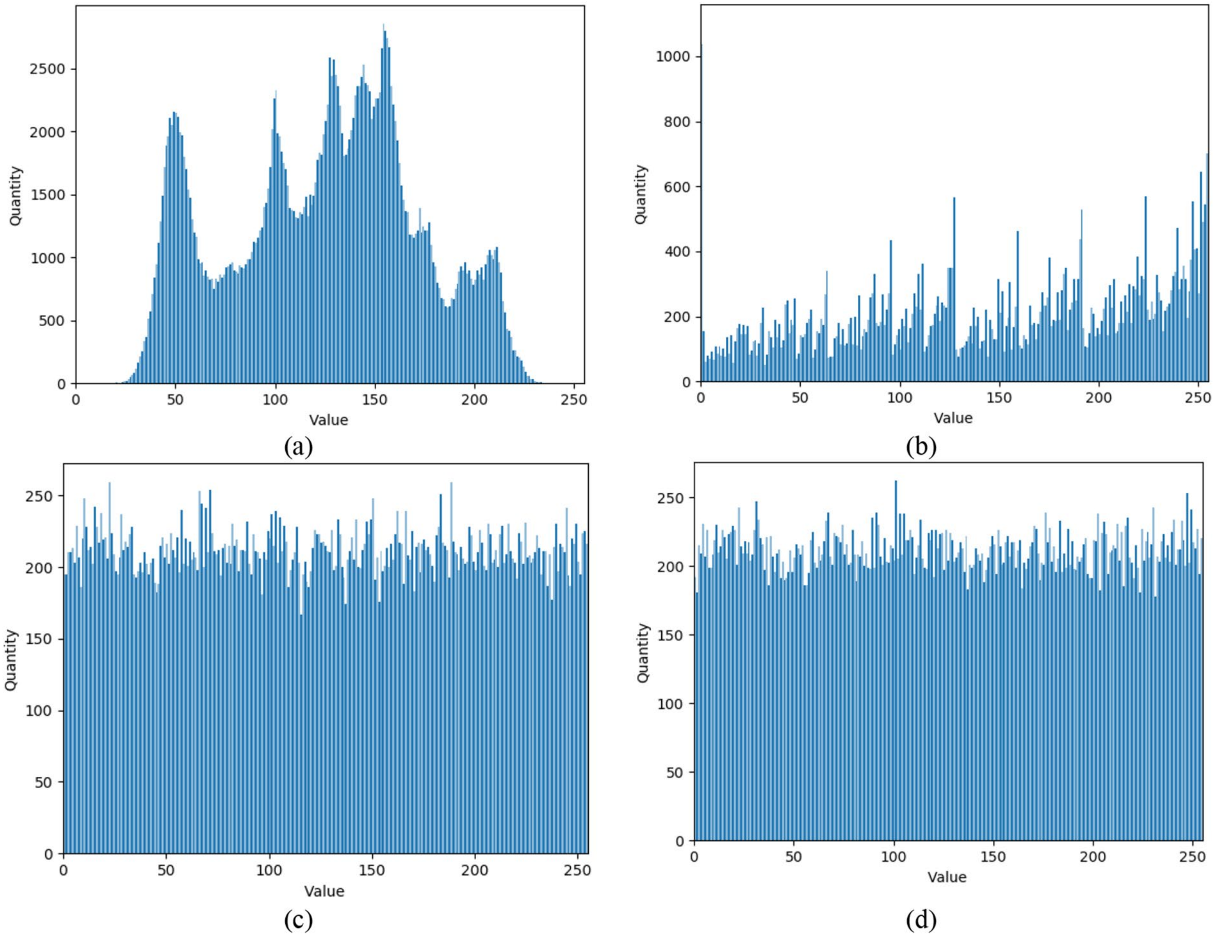
4.3 Connected component analysis(CCA)

Generally, for a plain image, there always exists a high correlation between adjacent pixels. Therefore, good encryption algorithms can result in smaller correlation values between adjacent pixels. We randomly select 3000 pixels from the information of plaint and encrypted images of image 1 in Figs. 18a–c. The correlation coefficients for the horizontal, vertical, and diagonal directions were calculated by using the following equations [33]:

$$CCA = \frac{\sum_{i=1}^{N}\left(x_i - \frac{1}{N}\sum_{i=1}^{N} x_i\right)\left(y_i - \frac{1}{N}\sum_{i=1}^{N} y_i\right)}{\sqrt{\sum_{i=1}^{N}\left(x_i - \frac{1}{N}\sum_{i=1}^{N} x_i\right)^2 \times \sum_{i=1}^{N}\left(y_i - \frac{1}{N}\sum_{i=1}^{N} y_i\right)^2}}, \tag{16}$$

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).

In Fig. 21, it can be seen that the original picture is highly correlated; while the picture converted to JPEG format has a more even distribution, but there are still clusters on the upper right. However, the encrypted images with one key and five keys, respectively, are distributed randomly and



(a)

(b)

(c)

(d)

Fig. 20 Histogram analysis (**a**) the original image(**b**) the original image with JPEG format (**c**) the cipher image by one key (static key) (**d**) the cipher image by five dynamic keys

Table 4 The variance of histogram

	The original image	The original image with JPEG format the cipher image	The cipher image by one key (static key)	The cipher image by five dynamic keys	[32]
Variance of histogram	6,443,335.6875	17,383.8787	232.2302	202.7537	1052.4

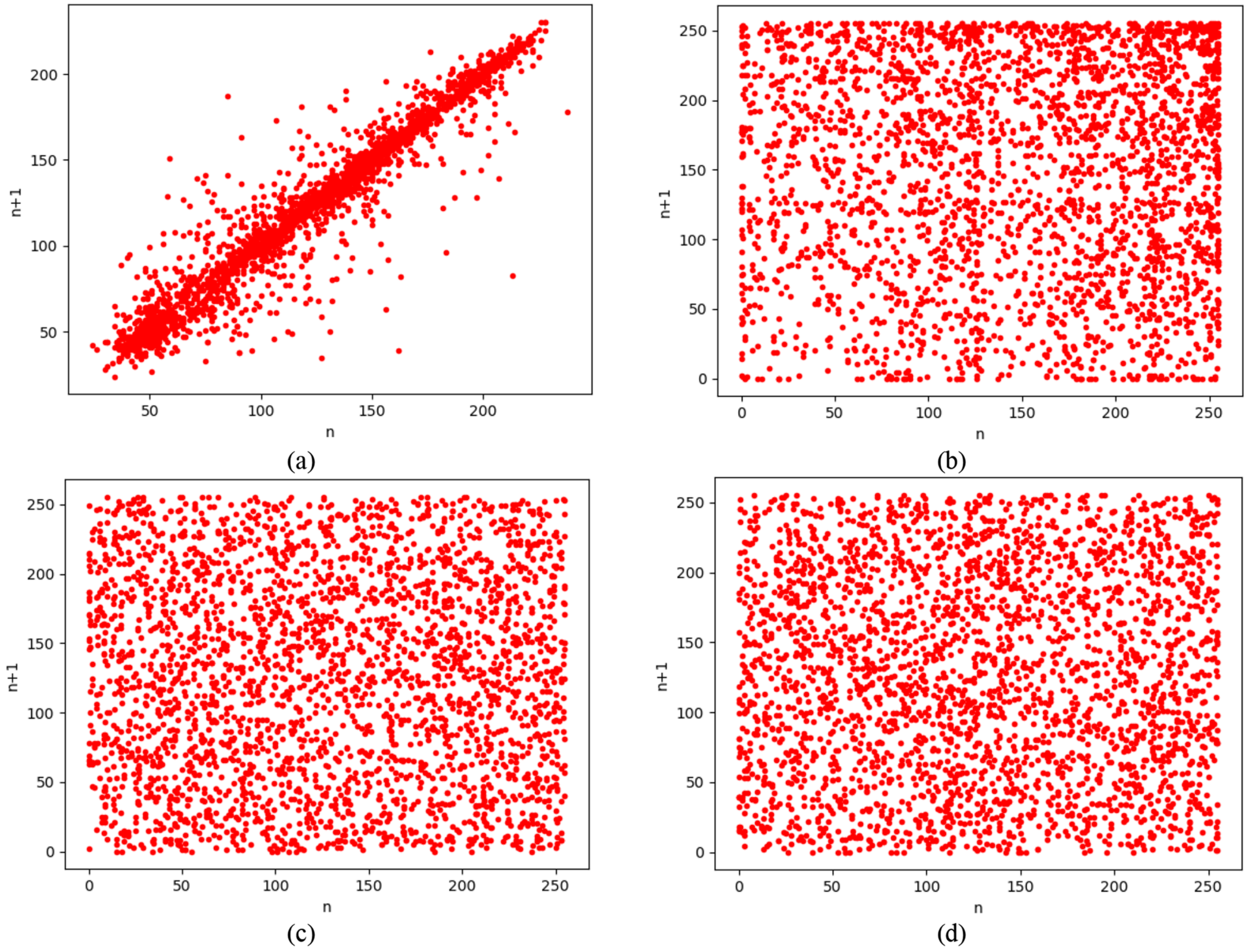
uniformly. Detailed reports of the connected component analysis are given in Tables 5 and 6. According to Tables 5 and 6, it could be seen that the encrypted image with the proposed five dynamic keys had the lowest pixel correlation which is superior to the results in the previous reports [32, 34, 35].

4.4 Information entropy analysis

Information entropy is a measure of uncertainty. We can judge the randomness degree of an event by calculating the entropy value. For the encrypted image, a large entropy value implies a better encryption effect. The calculation method is given as follows [36]:

$$H = -\sum_{i=1}^{255} p_i \log_2 p_i \tag{17}$$

where p_i is the frequency of each greyscale For a grayscale image, the pixel has a data field of [0, 255] and the maximum value of IE will be 8. We use the pictures in Fig. 18a-c for analysis. The calculation results are shown in Table 7. The results indicate that the original image has the smallest



(a)

(b)

(c)

(d)

Fig. 21 Connected component analysis (**a**) the original image (**b**) the original image with JPEG format (**c**) the cipher image by one key (static key) (**d**) the cipher image by five keys

Table 5 Connected component analysis (N = 3000)

N	3000				
Test image	The original image	The original image with JPEG format	The cipher image by one key (static key)	The cipher image by five keys	[32]
Horizontal	0.9699	−0.0564	0.0085	0.0061	−0.0685
Vertical	0.9864	0.0130	0.0033	0.0021	0.0857
Diagonal	0.9567	−0.0157	−0.0279	−0.0045	0.0059
Absolute sum	2.913	0.0851	0.0397	0.0127	0.1601

entropy and the encrypted image is with the maximum value of 7.9911 by five keys which means that the encryption effect is good. Therefore, it could be concluded that the encrypted image possessed true random signal property and the proposed algorithm could resist entropy attacks.

Table 6 Connected component analysis (N = 10,000, 15,000)

N	10,000		15,000	
Test image	The cipher image by five keys	[34]	The cipher image by five dynamic keys	[35]
Horizontal	0.0031	0.0067	0.0033	0.0846
Vertical	0.0077	0.0172	-0.0044	0.0583
Diagonal	0.0004	0.0147	-0.0083	0.0931
Absolute sum	0.0112	0.0386	0.0160	0.236

4.5 Keyspace analysis

In this paper, we introduce the discrete 4D LS hyper-chaotic system with four initial state variables. Therefore, the keyspace is 10^{20} when the calculation accuracy is 10^{-5}. Furthermore, Since the dynamic key generator is integrated with the SHA3-256 algorithm with keyspace 2^{256}, the keyspace of dynamic keys generated with our method is $S = 10^{20} \times 2^{256} = 1.158 \times 10^{97}$ which is large enough to resist the brute-force attack.

4.6 NPCR and UACI analysis

NPCR (Number of pixels change rate) and UACI (Unified average changing intensity) are often used to analyze the sensitivity of a plain image so that differential attacks can be resisted [37]. The formulas are defined as follows:

$$\mathrm{D}(i,j) = \begin{cases} 0, if \quad C_1(i,j)=C_2(i,j) \\ 1, if \quad C_1(i,j) \neq C_2(i,j) \end{cases} \tag{18}$$

$$NPCR = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \times 100\% \tag{19}$$

$$UACI = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{\left|C_1(i,j) - C_2(i,j)\right|}{255} \times 100\% \tag{20}$$

where M and N are the width and height of encrypted images. $C_1(i,j)$ and $C_2(i,j)$ are, respectively, the pixels with the location (i,j) of two ciphered images C_1 and C_2. To perform NPCR and UACI tests, we continuously captured 12 images as shown in Fig. 19. After converting to JPEG format, each image randomly selected a pixel to increase or decrease by 1 and the original images and the slightly changed images can be encrypted with 5 dynamic keys proposed in the paper for testing. The results are shown in Table 8. According to the test results in Table 8, the average NPCR test is 99.6138%, while the UACI is 33.4700%. Furthermore, a comparison of NPCR and UACI tests of the proposed method and the others is given in Table 9. The comparison results reveal that the method proposed in this paper has better results in NPCR and UACI tests.

Table 7 Information entropy analysis

	The original image	The original image with JPEG format	The cipher image by one key (static key)	The cipher image by five dynamics keys
Entropy of picture	7.0004	7.7969	7.9906	7.9911

s

Table 8 NPCR and UACI analysis of Real-time captured images

Real-time captured images	NPCR (%)	UACI (%)
Image 1	99.6178	33.4323
Image 2	99.6123	33.4789
Image 3	99.6021	33.4996
Image 4	99.6160	33.5292
Image 5	99.6160	33.3588
Image 6	99.6177	33.6297
Image 7	99.6059	33.5444
Image 8	99.6121	33.2078
Image 9	99.6032	33.3816
Image 10	99.5968	33.4566
Image 11	99.6672	33.5706
Image 12	99.5979	33.5512
Total average	99.6138	33.4700

Table 9 Comparison of NPCR and UACI criteria of proposed method and the others

Test methods	NPCR (%)	UACI (%)
Proposed method	99.6138	33.4700
Ferdus et.al. [37]	99.5499	26.5199
Lin and Wu [38]	99.6096	33.4673
Gupta et al. [39]	99.5899	28.4899
Liu et al. [40]	99.6097	33.4557
Zheng et al. [41]	99.5969	33.4599
Patro et al. [42]	99.6091	33.4914
Bisht et al. [43]	99.7024	27.9796

5 Conclusions

In this paper, a novel design of random number generators has been firstly proposed by integrating the chaos random property and the avalanche effect of SHA3-256. Then by the design of the synchronization controller of master–slave hyper-chaotic systems, the dynamic random keys can simultaneously be generated at both transmitter and receiver. Continuously, an improved AES CFB encryption algorithm with dynamic random keys is realized to encrypt /decrypt streaming audio/video data. In addition, several tests and analyses, such as visual effect, statistical analysis, histogram, information entropy analysis, keyspace analysis were all performed to show the capability and security of the improved chaos-based AES CFB algorithm.

Funding This study was funded by Ministry of Science and Technology, Taiwan (Grant nos. MOST-109-2221-E-167 -017 and 110-2218-E-006 -014 -MBK)

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Saha, R., Geetha, G., Kumar, G., Kim, T.H.: RK-AES: An improved version of AES using a new key generation process with random keys. Secur. Commun. Netw. **2018**, 11 (2018)
2. Arute, F., Arya, K., Martinis, J.M.: Quantum supremacy using a programmable superconducting processor. Nature **574**: 505–510. (2019). https://www.nature.com/articles/s41586-019-1666-5
3. Rössler, O.E.: An equation for hyperchaos. Phys. Lett. A **71**(2–3), 155–157 (1979)
4. Kim, S., Lee, B., Kim, D.H.: Experiments on chaos synchronization in two separate erbium-doped fiber lasers. IEEE Photonics Technol. Lett. **13**(4), 290–292 (2001)
5. Liu, S., Jiang, N., Zhao, A., Zhang, Y., Qiu, K.: Secure optical communication based on cluster chaos synchronization in semiconductor Lasers Network. IEEE Access **8**, 11872–11879 (2020)
6. Jun, B., Kocher, P.: Intel random number generator. Cryptography Research Inc. white paper, Cryptography Research, Inc. and Intel Corporation,State of California, USA (1999)
7. Kadam, M., Siddamal, S.V.: Annigeri S Design and implementation of chaotic non-deterministic random seed-based hybrid true random number generator. In: 2020 24th International Symposium on VLSI Design and Test (VDAT). (2020)
8. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., Vo, S.: A statistical test suite for random and pseudorandom number generators for cryptographic applications, pp 800–822. NIST Special Publication, USA (2010)
9. Ahmed, R.E.: Efficient pseudo-random number generators for wireless sensor networks. In: 2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS). (2017). https://doi.org/10.1109/MWSCAS.2016.7869989
10. Abutaha, M., Assad, S.E., Jallouli, O., Queudet, A., Deforges, O.: Design of a pseudo-chaotic number generator as a random number generator. In: 2016 International Conference on Communications (COMM). (2016). https://doi.org/10.1109/ICComm.2016.7528291
11. Ott, E., Grebogi, C., Yorke, J.A.: Controlling chaos. Phys Rev Lett **64**(11), 1196–1199 (1990)
12. Pecora, L.M., Carroll, T.L.: Synchronization in chaotic systems. Phys Rev Lett **64**(8), 821–824 (1990)
13. Li, Y., Zhao, Y., Yao, Z.A.: Chaotic control andgeneralized synchronization for a hyperchaotic lorenz-stenflo system. Abstr. Appl. Anal. (2013). https://doi.org/10.1155/2013/515106
14. Kai, G., Zhang, W., Wei, Z.C., Wang, J.F., Akgul, A.: Hopf bifurcation, positively invariant set, and physical realization of a New four-dimensional hyperchaotic financial system. Math. Probl. Eng. (2013). https://doi.org/10.1155/2017/2490580
15. Daryabor, A., Momeni, H.R.: A sliding mode observer approach to chaos synchronization. In: 2008 International Conference on Control, Automation and Systems. (2008). https://doi.org/10.1109/ICCAS.2008.4694492
16. Panikhom, S.: Implementation of chaos control in Chua's circuit via sliding mode control. 2017 International Electrical Engineering Congress (iEECON) 2017. (2017). https://doi.org/10.1109/IEECON.2017.8075916
17. Bryson, A.E.: Optimal control-1950 to 1985. IEEE Control Syst. Mag. **16**(3), 26–33 (1996)
18. Chen, G.: A simple adaptive feedback control method for chaos and hyper-chaos control. Appl. Math. Comput. **217**(17), 7258–7264 (2011)
19. Pai, M.C.: Global synchronization of uncertain chaotic systems via discrete-time sliding mode control. Appl. Math. Comput. **227**, 663–671 (2014)
20. Kuo, C.L.: Design of a fuzzy sliding-mode synchronization controller for two different chaos systems. Comput. Math. Appl. **61**(8), 2090–2095 (2011)
21. Xavier, J.C., Rech, P.C.: Regular and chaotic dynamics of the Lorenz-Stenflo system. Int.l Journal of Bifurc. Chaos **20**(1), 145–152 (2010)
22. Chen, Y.M., Liang, H.H.: Zero-zero-Hopf bifurcation and ultimate bound estimation of a generalized Lorenz-Stenflo hyperchaotic system. Math. Method Appl. Sci **40**, 3424–3432 (2017)
23. Young, K.D., Utkin, V.I., Ozguner, U.: A control engineer's guide to sliding mode control. IEEE Trans. Control Syst. Technol. **7**(3), 328–342 (1999)
24. Dworkin, M.J.: SHA-3 standard: permutation-based hash and extendable-output functions. FIPS PUB 202, USA (2015)
25. Dworkin, M.J., Barker, E.B., Nechvatal, J.R., Foti, J., Bassham, L.E., Roback, E., Dray, J.F.J.: Announcing the advanced encryption standard (AES). NIST FIPS 197. (2001). https://www.nist.gov/publications/advanced-encryption-standard-aes
26. Samet, H., Tamminen, M.: Efficient component labeling of images of arbitrary dimension represented by linear bintrees. IEEE Trans. Pattern Anal. Mach. Intell. **10**(4), 579–586 (1988)
27. Dillencourt, M.B., Samet, H., Tamminen, M.: A general approach to connected-component labeling for arbitrary image representations. J. ACM **39**(2), 253–280 (1992)
28. Wu, W., Huang, Y., Kurachi, R., Zeng, G., Xie, G., Li, R., Li, K.: Sliding window optimized information entropy analysis method for intrusion detection on in-vehicle networks. IEEE Access **6**, 45233–45245 (2018)
29. Ozdemir, A., Pehlivan, I., Akgul, A., Guleryuz, E.: A strange novel chaotic system with fully golden proportion equilibria and its mobile microcomputer-based RNG application. Chin. J. Phys. **56**(6), 2852–2864 (2018)
30. Lai, Q., Wan, Z., Akgul, A., Boyraz, O.F., Yildiz, M.Z.: Design and implementation of a new memristive chaotic system with application in touchless fingerprint encryption. Chin. J. Phys. **67**, 615–630 (2020)
31. Zhang, Y.Q., Wang, X.Y.: A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice. Inf. Sci. **273**, 329–351 (2014)

32. Tang, Z., Yang, Y., Xu, S., Yu, C., Zhang, X.: Image encryption with double spiral scans and chaotic maps. Security and Communication Networks **2019**, 15 (2019)
33. Zhang, W., Wong, K.W., Yu, H., Zhu, Z.L.: An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion. Commun. Nonlinear Sci. Numer. Simul. **18**(8), 2066–2080 (2013)
34. Lin, R.M., Ng, T.Y.: Secure image encryption based on an ideal new nonlinear discrete dynamical system. Math. Probl. Eng. **2018**, 12 (2018)
35. Lin, J., Luo, Y., Liu, J., Bi, J., Qiu, S., Cen, M., Liao, Z.: An image compression-encryption algorithm based on cellular neural network and compressive sensing. In: 2018 3rd IEEE International Conference on Image, Vision and Computing (ICIVC). (2018).
36. Tsai, D.Y., Lee, Y., Matsuyama, E.: Information entropy measure for evaluation of image quality. J. Digit. Imaging **21**(3), 338–347 (2008)
37. Ferdush, J., Begum, M., Uddin, M.S.: Chaotic lightweight cryptosystem for image encryption. Adv. Multimed. **2021**, 16 (2021)
38. Lin, C.Y., Wu, J.L.: Cryptanalysis and improvement of a chaotic map-based image encryption system using both plaintext related permutation and diffusion. Entropy **22**(5), 589 (2020)
39. Gupta, M., Gupta, K.K., Shukla, P.K.: Session key based fast, secure and lightweight image encryption algorithm. Multimed. Tools Appl.s **80**, 10391–10416 (2020)
40. Liu, H., Zhao, B., Zou, J., Huang, L., Liu, Y.: A lightweight image encryption algorithm based on message passing and chaotic map. Secur. Commun. Netw. **2020**, 12 (2020)
41. Zheng, J., Luo, Z., Tang, Z.: An image encryption algorithm based on multichaotic system and DNA coding. Discret. Dyn. Nat. Soc. **2020**, 16 (2020)
42. Patro, K.A.K., Acharya, B., Nath, V.: A secure multi-stage one-round bit-plane permutation operation based chaotic image encryption. Microsyst. Technol. **25**(6), 2331–2338 (2019)
43. Bisht, A., Dua, M., Dua, S., Jaroli, P.: A color image encryption technique based on bit-level permutation and alternate logistic-maps. J. Intell. Syst. **342**, 1–15 (2019)