Sensors and Materials, Vol. 37, No. 7 (2025) 2831–2850
MYU Tokyo

S & M 4090

Design of Image Cryptosystems Using Dynamic Synchronized Random Keys Based on Chaotic Extremum Coding

Chih-Hsueh Lin,[1,2] Guo-Hsin Hu,[3] Shunsuke Araki,[4]
Hong-Lin Mu,[5] and Jun-Juh Yan[5*]

[1]Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan
[2]Artificial Intelligence Innovation Institute, Institute for Information Industry, Taipei, Taiwan
[3]Department of Industrial Upgrading Service, Metal Industries Research & Development Centre, Kaohsiung, Taiwan
[4]Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology 680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan
[5]Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung, Taiwan

(Received December 27, 2024; accepted April 21, 2025)

Keywords: cryptosystem, synchronization control, extremum coding, chaos, image encryption

In this paper, a cryptosystem characterized by chaotic extremum coding and dynamic synchronized random keys is newly proposed. First, a novel chaotic extremum coding algorithm is developed to generate dynamic random numbers with good randomness quality. Meanwhile, the synchronization controllers are recommended to guarantee that the transmitter and receiver can simultaneously generate the same dynamic chaotic extremum-coded random keys (DCECRKs) for encryption and decryption, respectively. The designed DCECRKs are dynamically updated and can solve problems of key distribution and storage in the traditional encryption algorithm. Then, we apply the proposed DCECRKs to encrypt/decrypt image data. We employ the Advanced Encryption Standard Cipher Feedback encryption algorithm, integrated with the DCECRKs, to perform image encryption and decryption. To emphasize the contribution of the proposed DCECRKs and cryptosystem, a series of tests were conducted, including statistical analysis, histogram evaluation, connected component analysis, information entropy measurement, correlation indices assessment, number of pixels change rate, and unified average changing intensity. These results were compared against findings from existing literature, demonstrating that the proposed cryptosystem achieves superior security in image encryption.

1. Introduction

With the maturity of the fifth generation (5G) mobile communication technology, the achievement and application of the Internet of Things have provided considerable convenience in information exchange. However, while enjoying the convenience of information exchange, it

*Corresponding author: e-mail: jjyan@ncut.edu.tw
https://doi.org/10.18494/SAM5544

ISSN 0914-4935
https://myukk.org/

comes with more risks and threats to information security. Personal private data may be intercepted during transmission. Therefore, data security is a very important issue.[1] To ensure information security, traditional symmetric encryption, such as Data Encryption Standard, Advanced Encryption Standard (AES), and Rivest Cipher5 (RC5), integrates the mathematical complexity of the encryption algorithm with a static key to encrypt/decrypt data to ensure data security.[2] The principle of the symmetric encryption algorithm is that a static key is used for the data encryption/decryption. However, the risk of the symmetric encryption algorithm lies in the storage and distribution of the static key. Once the static keys are hacked, all the information will be completely exposed. In addition, how to ensure the security and uniqueness of the static keys in the process of distributing the keys for many users is a very important issue. Another encryption method is the asymmetric key algorithm, which is also known as the public key algorithm. It requires a public key for encryption and a private key for decryption, and the public key only encrypts plaintext function. Therefore, compared with the symmetric algorithm, the asymmetric encryption algorithm is more secure, but the asymmetric key algorithm still has the risk of private key storage, and although it is more secure than the symmetric key algorithm, its encryption and decryption efficiency is lower than that of symmetric encryption. However, according to the literature,[3,4] the improvement of computer processing speed and the maturity of quantum computer technology will make it possible to realize Shor or Grover's quantum calculation algorithm to solve the mathematical problems of these ciphers and the current cryptosystems will be at risk.[5,6] Therefore, the design of dynamically updated keys with high randomness will be useful to solve this problem. However, concerning the traditional encryption algorithms, it is difficult to dynamically update the keys and ensure the security of the keys during transmission.

The chaotic system is a high-complexity nonlinear system, and its dynamic state responses are quasi-random and aperiodic. Owing to the high sensitivity of chaotic motion to initial conditions (often referred to as the butterfly effect) and the presence of strange attractors,[7–10] chaotic systems have become extensively utilized in the design of data encryption methods.[11–17] The random number generators designed on the basis of chaos theory have a butterfly effect and unpredictable characteristics. Researchers had applied this feature to design dynamic keys for encryption algorithms, but the design of chaos synchronization control had not been considered in their studies.[13,14] Therefore, the cryptosystems must have the ensuing restrictions of the same initial conditions for master and slave chaotic systems. However, if the cryptosystem is suddenly disturbed and results in a very small state difference, the butterfly effect in chaotic systems causes the dynamic keys at the transmitter and receiver to become entirely distinct, leading to the breakdown of these cryptosystems. To overcome this challenge, chaotic cryptography researchers have designed synchronization controllers that effectively counteract the butterfly effect between master–slave chaotic systems, allowing for the generation of identical dynamic keys for both encryption and decryption.[15–17] However, the random number generation mechanisms[15–17] are simple, and the quality of the dynamic random number can still be improved.

Motivated by the aforesaid, in this work, we will design a new chaotic extremum-coded random key generator and establish a new cryptosystem to ensure data security. First, a novel

chaotic extremum coding algorithm, which will be thoroughly described later, is developed to generate dynamic random numbers with good randomness quality. Meanwhile, the synchronization controllers are derived to guarantee that the transmitter and receiver can simultaneously generate the same DCECRKs for encryption and decryption, respectively. After completing the design of the synchronized DCECRKs, we can realize the dynamic cryptosystem for image encryption/decryption. In this cryptosystem, the symmetric encryption algorithm AES CFB[18] is integrated with the proposed DCECRKs to dynamically encrypt/decrypt the images. Finally, to highlight the performance and contribution of the extremum coding approach, many tests including the National Institute of Standards and Technology (NIST) test,[19] the statistical analysis, histogram, connected component analysis (CCA), IE, correlation indexes, number of pixels change rate (NPCR), and unified average changing intensity (UACI) were conducted and the results were compared with those in the literature.

The structure of this paper is as follows. In Sect. 2, we describe the synchronization control of discrete chaotic systems. The extremum coding and the proposed DCECRKs are formulated in Sect. 3. In Sect. 4, the proposed cryptosystem with DCECRKs is applied to image encryption/decryption. Many tests and comparisons are given to highlight the contributions of the extremum coding approach. Finally, brief conclusions are included in Sect. 5.

2. Synchronization in Discrete Master and Slave Chaotic Systems

In this cryptosystem design, instead of the continuous chaotic system, we will use the discrete chaotic systems for research. The main reason is that when the continuous chaotic system is implemented in the digital microcontroller, the synchronization control performance is often affected by the delay of the network environment. The time delay and variance are difficult to grasp; therefore, it might produce unpredictable results and result in the failure of synchronization. In the following section, we first demonstrate the process of discretizing a continuous chaotic system while preserving its chaotic behavior. The dynamic equation of a continuous nonlinear chaotic system can be expressed as

$$\dot{x}(t) = Ax(t) + Bg\left(x(t)\right), \tag{1}$$

where $x(t) \in R^{n\times 1}$ is the state vector, $A \in R^{n\times n}$ and $B \in R^{n\times m}$ are system matrices, and $g(x(t))$ denotes the nonlinear vector. The discrete counterpart of system (1) can thus be expressed as

$$x\left((k+1)T\right) = Gx\left(kT\right) + Hg\left(x\left(kT\right)\right), \tag{2}$$

where $G = e^{AT}$, T is the sampling time, and $H = \left[G - I_n\right]A^{-1}B$.[20] In the following, we present a four-dimensional chaotic system (FDCS) for our design; however, the developed results can be easily adapted to other chaotic systems. The new FDCS is governed by the following state equations:[21]

$$\begin{aligned}\dot{x}(t) &= -ax(t) + y(t)z(t),\\ \dot{y}(t) &= by(t) - x(t)z(t),\\ \dot{z}(t) &= -cz(t) + y(t)w(t),\\ \dot{w}(t) &= -dw(t) - x(t)y(t).\end{aligned} \tag{3}$$

Clearly, Eq. (3) with the parameters $(a,b,c,d) = (4,1,2,0.5)$ can be rearranged into the form of Eq. (1), with the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ presented below.

$$\boldsymbol{A} = \begin{bmatrix} -4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -0.5 \end{bmatrix}; \boldsymbol{B} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

With a sampling time of $T = 0.001$, by using $G = e^{AT}$ and $H = \left[G - I_n\right]A^{-1}B$, the corresponding discrete version of system (3) can be obtained as

$$\begin{aligned}x(k+1) &= 0.996x(k) + 0.001y(k)z(k),\\ y(k+1) &= 1.001y(k) - 0.001x(k)z(k),\\ z(k+1) &= 0.998z(k) + 0.001y(k)w(k),\\ w(k+1) &= 0.995w(k) - 0.001x(k)y(k).\end{aligned} \tag{4}$$

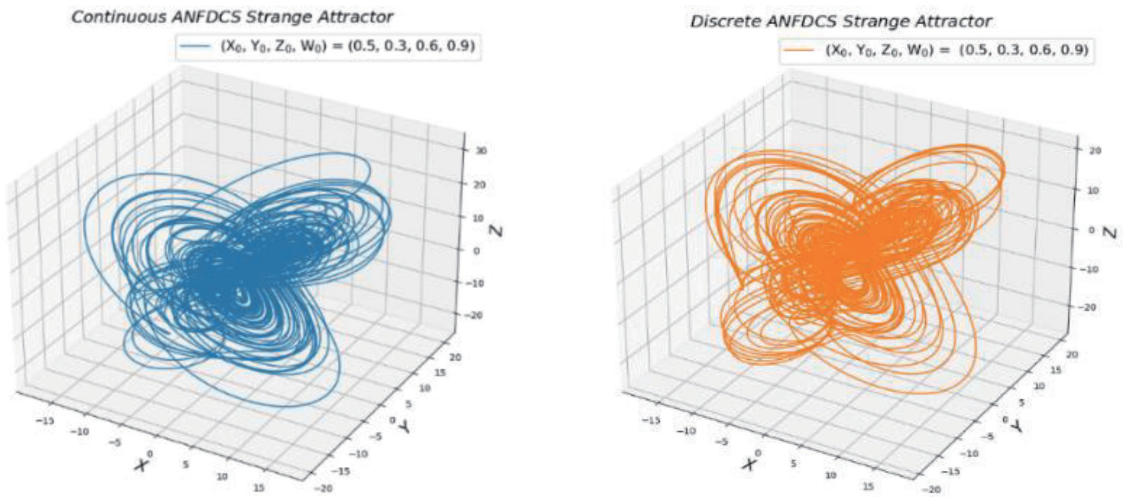
We simulate the strange attractor responses of both the continuous FDCS (3) and the discrete system (4) under identical initial conditions. The simulation results, presented in Fig. 1, demonstrate that the discretization of the continuous FDCS (3) retains the chaotic behavior in the corresponding discrete system (4).

In the following discussion, we will design the synchronization controller for the discretized chaotic systems, which will be used in the extremum coding architecture. The following describes the design of synchronization controllers for master–slave FDCS (4) and two-dimensional triangle function discrete chaotic maps (2D-TFCDM).[22]

FDCS synchronization control: In this master–slave discrete chaotic synchronization architecture, the master–slave systems are designed as follows.

Master FDCS system:

$$\begin{aligned}x_1(k+1) &= 0.996x_1(k) + 0.001x_2(k)x_3(k)\\ x_2(k+1) &= 1.001x_2(k) - 0.001x_1(k)x_3(k)\\ x_3(k+1) &= 0.998x_3(k) + 0.001x_2(k)x_4(k)\\ x_4(k+1) &= 0.995x_4(k) - 0.001x_1(k)x_2(k)\end{aligned} \tag{5}$$



(a) (b)

Fig. 1. (Color online) Strange attractors: (a) for the continuous FDCS (3) and (b) for the discrete chaotic system (4).

Slave FDCS system:

$$\begin{aligned} y_1(k+1) &= 0.996y_1(k) + 0.001y_2(k)y_3(k) + u_1(k) \\ y_2(k+1) &= 1.001y_2(k) - 0.001y_1(k)y_3(k) + u_2(k) \\ y_3(k+1) &= 0.998y_3(k) + 0.001y_2(k)y_4(k) \\ y_4(k+1) &= 0.995y_4(k) - 0.001y_1(k)y_2(k) \end{aligned} \tag{6}$$

Here, $u_1(k)$ and $u_2(k)$ are the synchronization control inputs. The state errors between the master and slave systems are defined as

$$e_i(k) = y_i(k) - x_i(k), i = 1,2,3,4. \tag{7}$$

Then the error dynamics between Eqs. (5) and (6) is obtained as

$$\begin{aligned} e_1(k+1) &= 0.996e_1(k) + 0.001\big(y_2(k)y_3(k) - x_2(k)x_3(k)\big) + u_1(k), \\ e_2(k+1) &= 1.001e_2(k) - 0.001\big(y_1(k)y_3(k) - x_1(k)x_3(k)\big) + u_2(k), \\ e_3(k+1) &= 0.998e_3(k) + 0.001\big(y_2(k)y_4(k) - x_2(k)x_4(k)\big), \\ e_4(k+1) &= 0.995e_4(k) - 0.001\big(y_1(k)y_2(k) - x_1(k)x_2(k)\big). \end{aligned} \tag{8}$$

In the following, we will refer to the design concept of the sliding mode control.[23] Following the design procedures, we first need to choose an appropriate sliding surface to ensure that when the error system (8) is in the sliding mode, the error states can converge to zero. Then, we need to design the control inputs to ensure the existence of the sliding mode. Therefore, the design of the synchronization controller can be completed in the following two steps.

Step 1: Design of an appropriate sliding surface: We choose the sliding surface $s(k)=\begin{bmatrix} e_1(k) & e_2(k) \end{bmatrix}^T$; then, we have

$$\begin{aligned} s(k+1) &= \begin{bmatrix} s_1(k+1) & s_2(k+1) \end{bmatrix}^T = \begin{bmatrix} e_1(k+1) & e_2(k+1) \end{bmatrix}^T \\ &= \begin{bmatrix} 0.996e_1(k)+0.001\left(y_2(k)y_3(k)-x_2(k)x_3(k)\right)+u_1(k) \\ 1.001e_2(k)-0.001\left(y_1(k)y_3(k)-x_1(k)x_3(k)\right)+u_2(k) \end{bmatrix}. \end{aligned} \tag{9}$$

In the following step 2, the control inputs $u_1(k)$ and $u_2(k)$ are designed to ensure that the error system [Eq. (8)] can enter the sliding mode (i.e., $s(k)=0$). Therefore, we can ensure $e_1(k)=e_2(k)=0$, which means the facts of $x_1(k)=y_1(k); x_2(k)=y_2(k)$. Thus, the last equation in Eq. (8) degenerates as $e_4(k+1)=0.9995e_4(k)$. Therefore, $e_4(k)$ can converge to zero and $y_4(k)=x_4(k)$. Consequently, the third equation in Eq. (8) degenerates as $e_3(k+1)=0.998e_3(k)$ since $y_2(k)=x_2(k)$ and $y_4(k)=x_4(k)$. Therefore, $e_3(k)$ can converge to zero and $y_3(k)=x_3(k)$. Thus, we conclude that the synchronization error $e_i(k)=y_i(k)-x_i(k), i=1,2,3,4$ can converge to zero and achieve complete synchronization. Next, it is still necessary to design the control inputs $u_1(k)$ and $u_2(k)$ to ensure the existence of the sliding mode ($s(k)=0$).

Step 2: Design of the control inputs: As mentioned in step 1, we must design the control inputs to ensure the existence of the sliding mode. To accomplish this goal, the control inputs $u_1(k)$ and $u_2(k)$ are designed as

$$u_1(k)=-\left(0.996e_1(k)+0.001\left(y_2(k)y_3(k)-x_2(k)x_3(k)\right)\right)+\alpha_1 s_1(k), \tag{10}$$

$$u_2(k)=-\left(1.001e_2(k)-0.001\left(y_1(k)y_3(k)-x_1(k)x_3(k)\right)\right)+\alpha_2 s_2(k), \tag{11}$$

where $|\alpha_1|<1; |\alpha_2|<1$.

It is obvious that when substituting Eqs. (10) and (11) into Eq. (9), we have

$$s_1(k+1)=\alpha_1 s_1(k); s_2(k+1)=\alpha_2 s_2(k). \tag{12}$$

Since $|\alpha_1|<1$ and $|\alpha_2|<1$ are selected, it is ensured that $s_1(k)=s_2(k)=0$ when $k\to\infty$. Furthermore, from Eq. (12), if $\alpha_1=\alpha_2=0$ is chosen, $s(k)$ as well as $e_1(k)$ and $e_2(k)$ will converge to zero using only one sampling time.

The following simulation will verify the correctness of the proposed FDCS synchronization. Let the parameters in Eqs. (10) and (11) be $\alpha_1=\alpha_2=0$ and the initial values of the master-slave system be $x_1(0)$=0.2; $x_2(0)$ = 0.5; $x_3(0)$ = 0.55; $x_4(0)$ = 0.74 and $y_1(0)$=0.87; $y_2(0)$ = 0.57; $y_3(0)$ = 0.87; $y_4(0)$ = 0.56. From the simulation results shown in Figs. 2 and 3, it is observed that $e_1(1)=e_2(1)=0$ [i.e., $x_1(1)=y_1(1)$ and $x_2(1)=y_2(1)$] when k = 1, and $e_3(k)$, and $e_4(k)$, also asymptotically converge to zero, as discussed above.

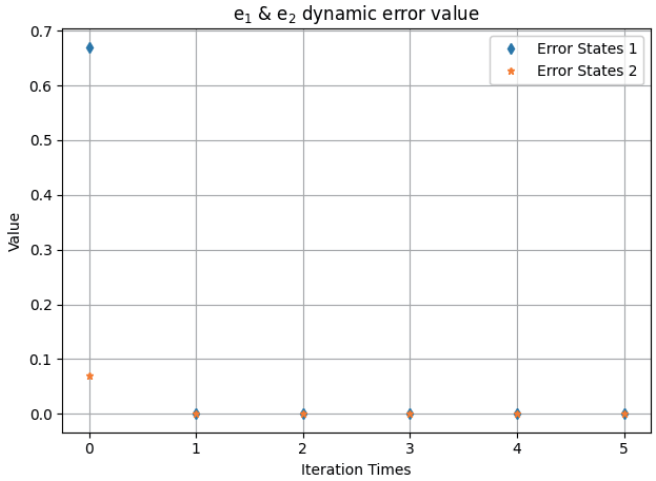


Fig. 2. (Color online) Instantaneous synchronization for e_1 and e_2.

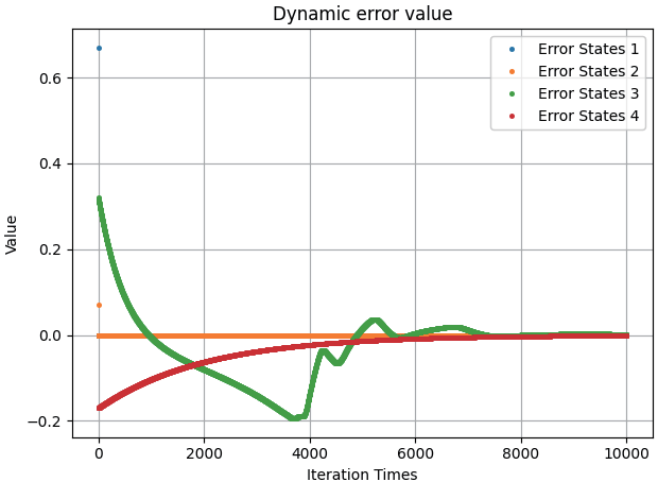


Fig. 3. (Color online) Error state responses.

2D-TFCDM synchronization control: Next, the synchronization control of 2D-TFCDM responsible for generating the AES chaotic dynamic keys is discussed. Because the inference is similar to the synchronization design of the above-mentioned FDCS, for simplification, only a brief description is given.

Master 2D-TFCDM system:

$$\begin{aligned} x_1(k+1) &= 6\cos\left(x_2(k)\right) \\ x_2(k+1) &= 0.3x_1(k) + 0.6x_2(k) \end{aligned} \tag{13}$$

Slave 2D-TFCDM system:

$$\begin{aligned} y_1(k+1) &= 6\cos\left(y_2(k)\right) \\ y_2(k+1) &= 0.3y_1(k) + 0.6y_2(k) + u(k) \end{aligned} \tag{14}$$

Let the error states be $e_i(k) = y_i(k) - x_i(k), i = 1,2$; the error dynamics can be obtained as

$$\begin{aligned} e_1(k+1) &= 6\left(\cos\left(y_2(k)\right) - \cos\left(x_2(k)\right)\right), \\ e_2(k+1) &= 0.3e_1(k) + 0.6e_2(k) + u(k). \end{aligned} \tag{15}$$

The control input $u(k)$ is designed as

$$u(k) = -\left(0.3e_1(k) + 0.6e_2(k)\right) + \alpha s(k), \tag{16}$$

where the sliding surface $s(k) = e_2(k)$, and $|\alpha| < 1$. By substituting Eq. (16) into Eq. (15), we have $s(k+1) = \alpha s(k)$ and the master-slave 2D-TFCDM chaotic systems can be completely synchronized, i.e., $y_i(k) = x_i(k), i = 1,2$. Furthermore, if $\alpha = 0$ is chosen, $s(k)$ as well as $e_2(k)$ will instantaneously converge to zero. $e_1(k)$ also converges to 0 after a sampling time following the convergence of $e_2(k)$ to 0. Let the initial condition of master and slave systems be $x_1(0) = 1.99, x_2(0) = -2.45$; $y_1(0) = -5.35, y_2(0) = 9.23$ and $\alpha = 0$. From the simulation results shown in Fig. 4, it is observed that $e_1(k)$ and $e_2(k)$ converge to zero as expected.

3. Extremum Coding and Design of DCECRKs

As far as cryptosystems are concerned, the keys for encryption are very important. To enhance the randomness of the dynamic chaos-based keys, we integrate chaos synchronization technology with extremum coding, leading to the development of synchronized DCECRKs. The architecture is shown in Fig. 5. Figure 5 includes N chaotic signal generators, a multiplexer, and a random extremum-coded sequence and a function of SHA256. The random extremum-coded sequence is the input of the multiplexer to randomly select different chaotic signals and then generate a fixed-length dynamic key through SHA256. Later, the quality of the keys obtained in Fig. 5 will be verified through many tests and comparisons with the results in the literature. The extremum coding controller in Fig. 5 is designed to formulate the extremum-coded sequence. As the extremum values are dynamically and randomly generated by the states of the FDCS chaotic systems, an extremum-coded sequence is created using extremum coding. We harness the unpredictable random state responses of FDCS chaotic systems to define the extremum coding rule, generating an unpredictable extremum-coded sequence. This sequence is subsequently used as the input for the multiplexer in Fig. 5. Therefore, by the multiplexer, one of the N chaotic signal generators in Fig. 5 can be randomly selected as the key generation according to n ($2^n \geq N$) least significant bits in the extremum-coded sequence. This design ensures that the extremum-coded sequence remains dynamic and random, with unpredictable switching timing, which effectively increases the difficulty of decryption. The proposed extremum coding can be

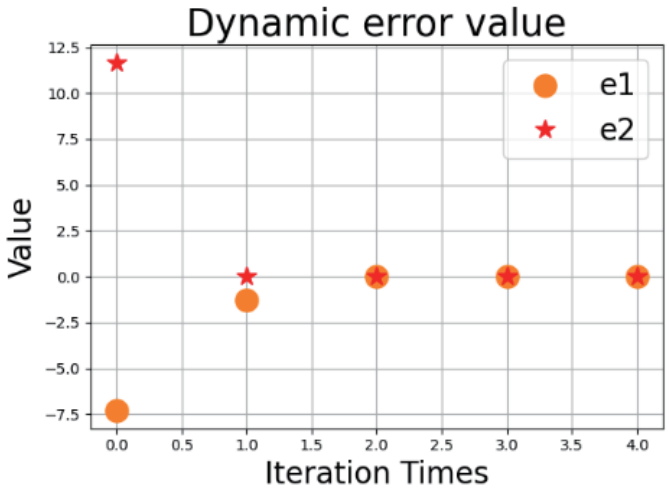


Fig. 4. (Color online) Instantaneous synchronization for e_1 and e_2.

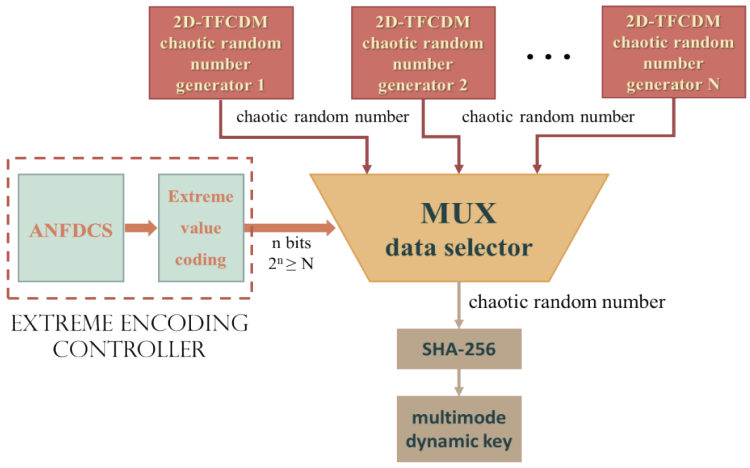


Fig. 5. (Color online) Architecture of dynamic chaotic extremum-coded random keys.

easily implemented on a microcontroller with simple programming. For a chaotic state response of FDCS, we record the peak values of the relative maxima, $P_i, i = 1,2,\ldots\infty$ and the trough values of the relative minima, $T_i, i = 1,2,\ldots\infty$, then compare the peaks or troughs, respectively. If $P_{i+1} > P_i$ or $T_{i+1} > T_i$, we store a 1; otherwise, we store a 0 as the new least significant bit (LSB) of the extremum-coded sequence. This extremum coding rule generates the subsequent sequence, with the flowchart for the generation process shown in Fig. 6.

By introducing the state response of the discrete FDCS (4) for performing the extremum coding in Fig. 6, the extremum-coded sequence with the initial sequence of 01 is [01101011000] as shown in Fig. 7.

In the following, we introduce FDCS (4) and four 2D-TFCDMs for performing the extremum coding and generating different chaotic signals, respectively. The initial values of four

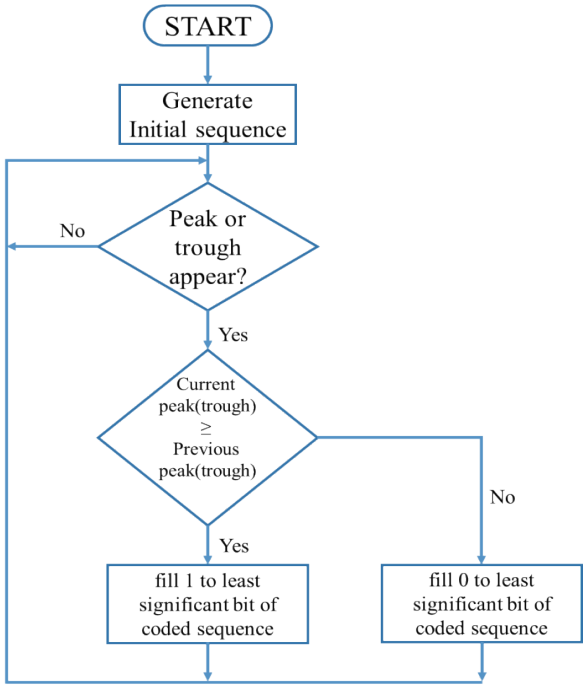


Fig. 6. (Color online) Flowchart for generating the extremum-coded sequence.

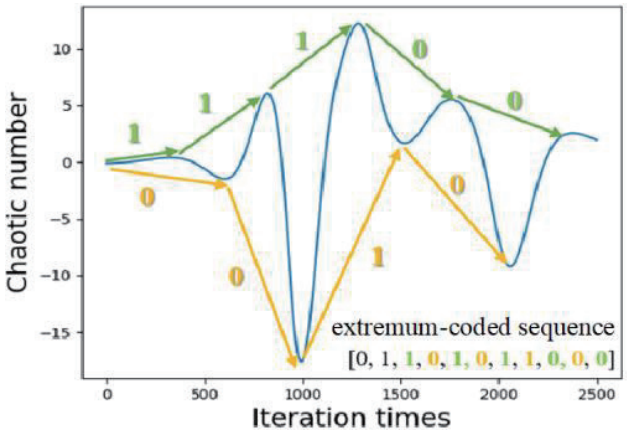


Fig. 7. (Color online) Extremum-coded sequence.

2D-TFCDMs are (0.31475, 0.23545), (0.35487, 0.12358), (0.12045, 0.57766), and (0.77855, 0.45784), respectively. Owing to the butterfly effect with different initial conditions, the given four 2D-TFCDMs will generate quite different chaotic signals. Figure 8 shows the output of the selected chaotic signal according to the extremum-coded sequence in Fig. 7. In this example, the chaotic random number generators N are 4 and the input bits n are 2 for the multiplexer.

From Fig. 8 above, at the beginning, 01 is the initial extremum-coded sequence, so the multiplexer selects the chaotic signal from the second chaotic signal generator 1. When the extremum-coded sequence is [01101011000], the lowest 2-bit is 00, which is the input of the

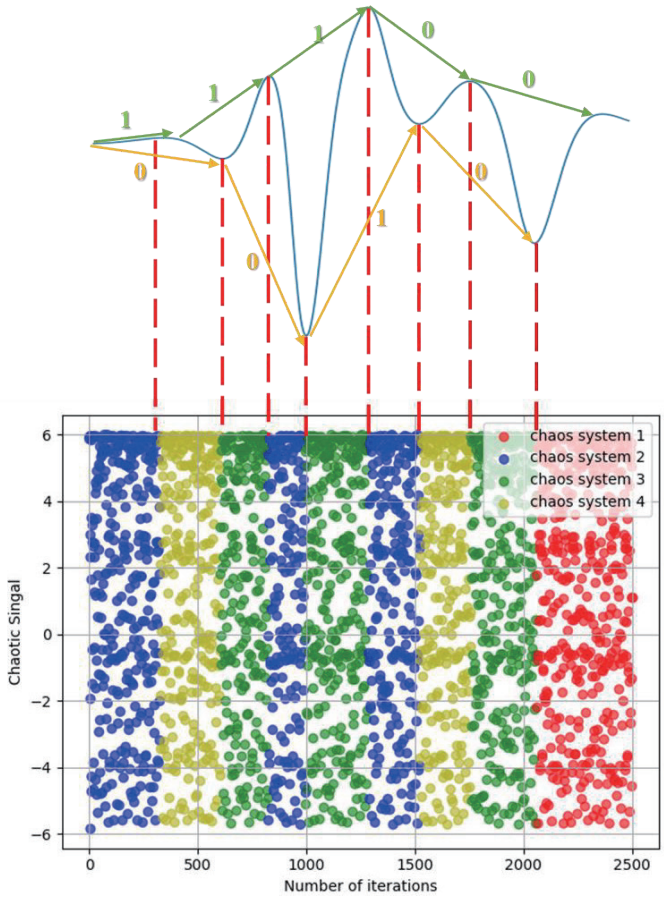


Fig. 8. (Color online) Output according to the extremum-coded sequence.

multiplexer, so it will select the chaotic signal from the first chaotic signal generator 0. From the above description, the use of extremum coding technology can allow the system to continuously and randomly switch between the different chaotic signal generators to improve the security of the proposed cryptosystem.

Design of DCECRKs and synchronization: From the above description, we clearly understand that through the extremum-coded sequence, one of the multi-mode chaotic systems can be dynamically chosen and can generate the required random inputs for the hash function (SHA256). Then, combined with the avalanche effect of the hash function (SHA256), the dynamic 256-bit key can be obtained. Meanwhile, the synchronization of master–slave FDCS and 2D-TFCDM chaotic systems is ensured as described in Sect. 2 to guarantee that the transmitter and receiver can simultaneously generate the same DCECRKs for encryption and decryption, respectively.

4. Security Analysis of DCECRKs and Image Cryptosystem

Security analysis of DCECRKs: The NIST test suite[19] is introduced to evaluate the randomness of the dynamic extremum-coded random keys. This test suite can be downloaded and installed from the official NIST website (NIST Special Publication (SP) 800-22 Revision 1a) and is a publicly available free resource. The NIST test comprises 15 items, and a test item is considered passed if its p-value exceeds 0.01. A large p-value signifies good randomness in the tested sequence. Four chaotic signal generators ($N = 4$) and two least significant bits ($n = 2$) in the extremum-coded sequence are used. Therefore, the four initial conditions of 2D-TFCDM systems are selected as $(x_{10}, y_{10}) = (-2.3022, 0.3475)$, $(x_{20}, y_{20}) = (-4.375, -1.4639)$, $(x_{30}, y_{30}) = (-1.153, 0.8237)$, and $(x_{40}, y_{40}) = (2.0142, 4.86)$; the initial condition of FDCS for generating the extremum-coded sequence is selected as (x_0, y_0, z_0, w_0) = (3.6612, 2.7801, 3.35, 4.9939). The states $x_{1i}, i = 1,2,3,4$ of 2D-TFCDM systems and x_1 of FDCS are used for generating the selected chaotic signals and extremum-coded sequence, respectively. To compare with the traditional single-mode key design, in the single-mode test, we disable the extremum coding and only select the chaotic state x_1 of a 2D-TFCDM for SHA256. Table 1 presents the test scores obtained for a stream length of 7360000 bits with a bit-stream count of 30. Table 1 shows that the test results of the DCECRK are better than those of the traditional single-mode design, and it can be verified that the randomness quality of the DCECRK obtained by the architecture in Fig. 5 is indeed considerably improved.

Next, we compare and analyze the proposed DCECRK with the recent literature regarding the design of random number generators. Under the same conditions, when $N = 10^6$ bits and a bit-stream count of 30, the comparison results are as shown in Table 2. It can be seen that the randomness quality of the DCECRK designed in this study is better than the results shown in the literature.[24–26]

Table 1
NIST SP 800-22 test for traditional chaotic dynamic key and DCECRK.

NIST SP 800-22	Traditional single-mode chaotic dynamic key	DCECRK (this paper)
	$N = 7360000$	
Frequency test	0.51213	0.99896
Frequency within block test	0.59681	0.86794
Runs test	0.84159	0.95136
Longest run ones in a block test	0.86588	0.69662
Binary matrix rank test	0.49798	0.12677
Discrete Fourier transform test	0.11090	0.64864
Non-overlapping template matching test	0.96853	0.99785
Overlapping template matching test	0.42117	0.90032
Maurer's universal statistical test	0.33197	0.43606
Linear complexity test	0.52604	0.01979
Serial test	0.57070	0.84609
Approximate entropy test	0.87608	0.95163
Cumulative sums test	0.79365	0.74903
Random excursions test	0.10057	0.12963
Random excursions variant test	0.12141	0.04386
Sum	**8.13547**	**9.36460**

Table 2
NIST test results (24–26) and this study.

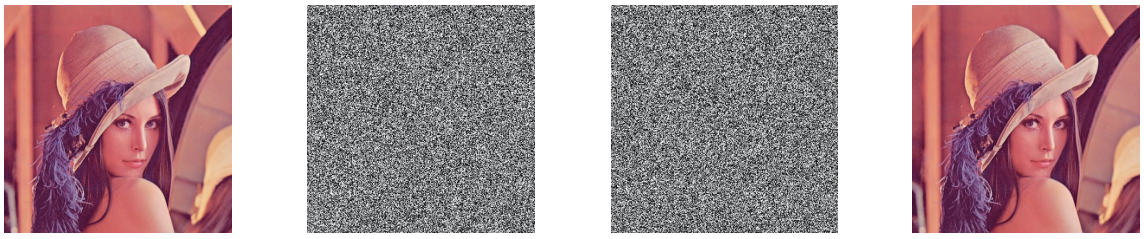
NIST SP 800-22	*P*-value ($N = 10^6$)			
	This paper	Charalampidis *et al.*[24]	Lin *et al.*[25]	Lin *et al.*[25]
Frequency test	0.3213	0.8831	0.4707	0.4707
Frequency within block test	0.3139	0.1153	0.6266	0.6266
Runs test	0.7582	0.8343	0.7230	0.7230
Longest run ones in a block test	0.5815	0.0269	0.2466	0.2466
Binary matrix rank test	0.6262	0.6786	0.5345	0.5345
Discrete Fourier transform test	0.8074	0.0909	0.3881	0.3881
Non-overlapping template matching test	0.9998	0.0235	0.9998	0.9998
Overlapping template matching test	0.3829	0.6993	0.6400	0.6400
Maurer's universal statistical test	0.7421	0.7981	0.6234	0.6234
Linear complexity test	0.5772	0.3351	0.6615	0.6615
Serial test	0.3761	0.8343	0.3579	0.3579
Approximate entropy test	0.5974	0.3838	0.3685	0.3685
Cumulative sums test	0.2740	0.1626	0.2687	0.2687
Random excursions test	0.3302	0.2622	0.3566	0.3566
Random excursions variant test	0.09210	0.0135	0.1852	0.1852
Sum	**7.7802**	**6.1415**	**7.4511**	**7.4511**

Security analysis of image cryptosystem using DCECRKs: After showing that the DCECRK design proposed in this paper can effectively improve the quality of random numbers, we further combine DCECRK with the AES CFB algorithm to encrypt/decrypt the images. Furthermore, to highlight the contribution of the proposed extremum coding and cryptosystem, many tests including the statistical analysis, histogram, connected component analysis, information entropy, correlation indexes, number of pixels change rate, and unified average changing intensity were conducted and the results were compared with those in the literature.

Visual effect of encrypted images: By using AES CFB with the proposed DCECRKs, Fig. 9(a) is encrypted and the results are shown in Figs. 9(b) and 9(c) with one static key and five dynamic keys, respectively. It can be observed that no obvious features exist in the cipher images and the decrypted image is shown in Fig. 9(d).

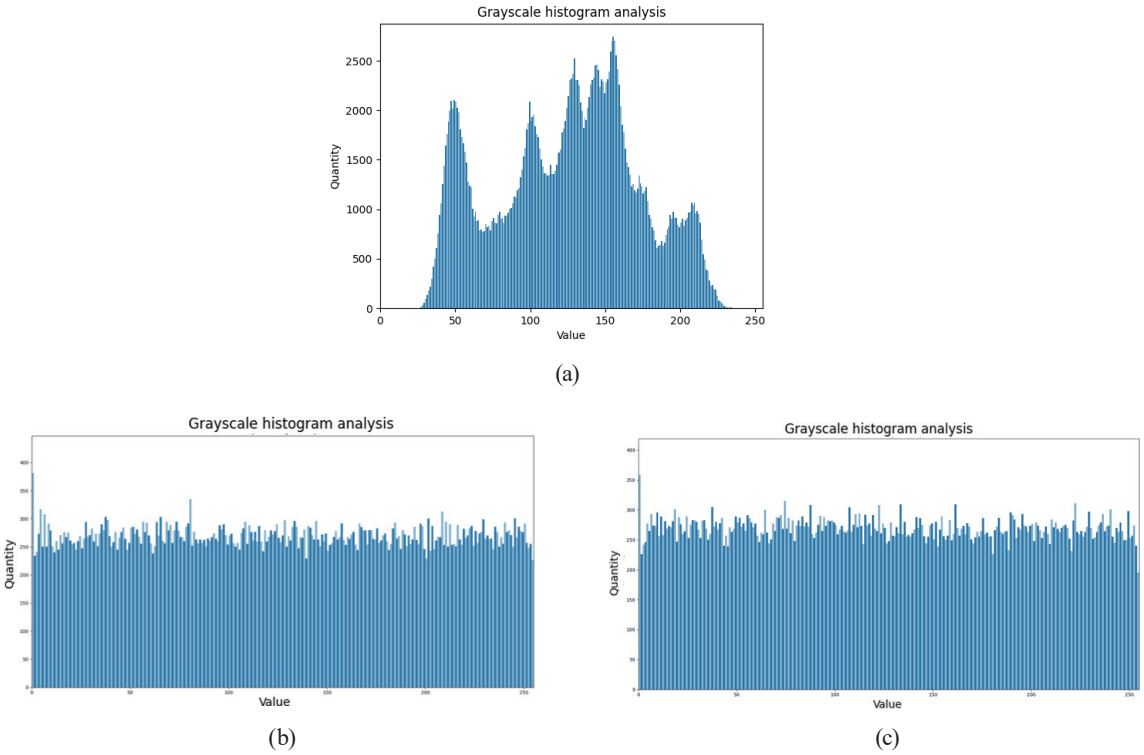
Histogram analysis: In image analysis, a histogram is the way to directly evaluate the effect of the cipher image. A cryptosystem with good performance can make the histogram of the encrypted image evenly distributed such that attackers cannot find obvious statistical properties from the encrypted image. An unencrypted image will have obvious peaks and troughs in the histogram as shown in Fig. 10(a), which is the histogram analysis of the Lena image in Fig. 9(a). Figures 10(b) and 10(c) are the histogram analyses for the encrypted images, respectively, using a static key and five dynamic keys. From Figs. 10(b) and 10(c), the histograms are close to the horizontal shape, and there are no large peaks or troughs. Therefore, it proves that the effect of encryption is effective.

CCA: In an unencrypted image, adjacent pixels will have a very high correlation. A superior image cryptosystem can reduce the correlation between adjacent pixels in the cipher image close to zero. By the following Eq. (17),[27] the CCA for the horizontal, vertical, and diagonal directions can be calculated.



(a) (b) (c) (d)

Fig. 9. (Color online) Visual effect of encrypted images.



(a)

(b) (c)

Fig. 10. (Color online) Histogram analysis: (a) original image, (b) cipher image by one key (static key), and (c) cipher image by five dynamic keys.

$$CCA = \frac{\sum_{i=1}^{N}\left(x_i - \frac{1}{N}\sum_{i=1}^{N}x_i\right)\left(y_i - \frac{1}{N}\sum_{i=1}^{N}y_i\right)}{\sqrt{\sum_{i=1}^{N}\left(x_i - \frac{1}{N}\sum_{i=1}^{N}x_i\right)^2 \times \sum_{i=1}^{N}\left(y_i - \frac{1}{N}\sum_{i=1}^{N}y_i\right)^2}} \tag{17}$$

Here, x_i and y_i are the pixel values of the two adjacent pixels and N is the number of (x_i, y_i). Three thousand pixels are randomly selected from the plaint and encrypted images, respectively, in Figs. 11(a)–11(c). Clearly, the original image [Fig. 11(a)] exhibits high correlation, whereas the

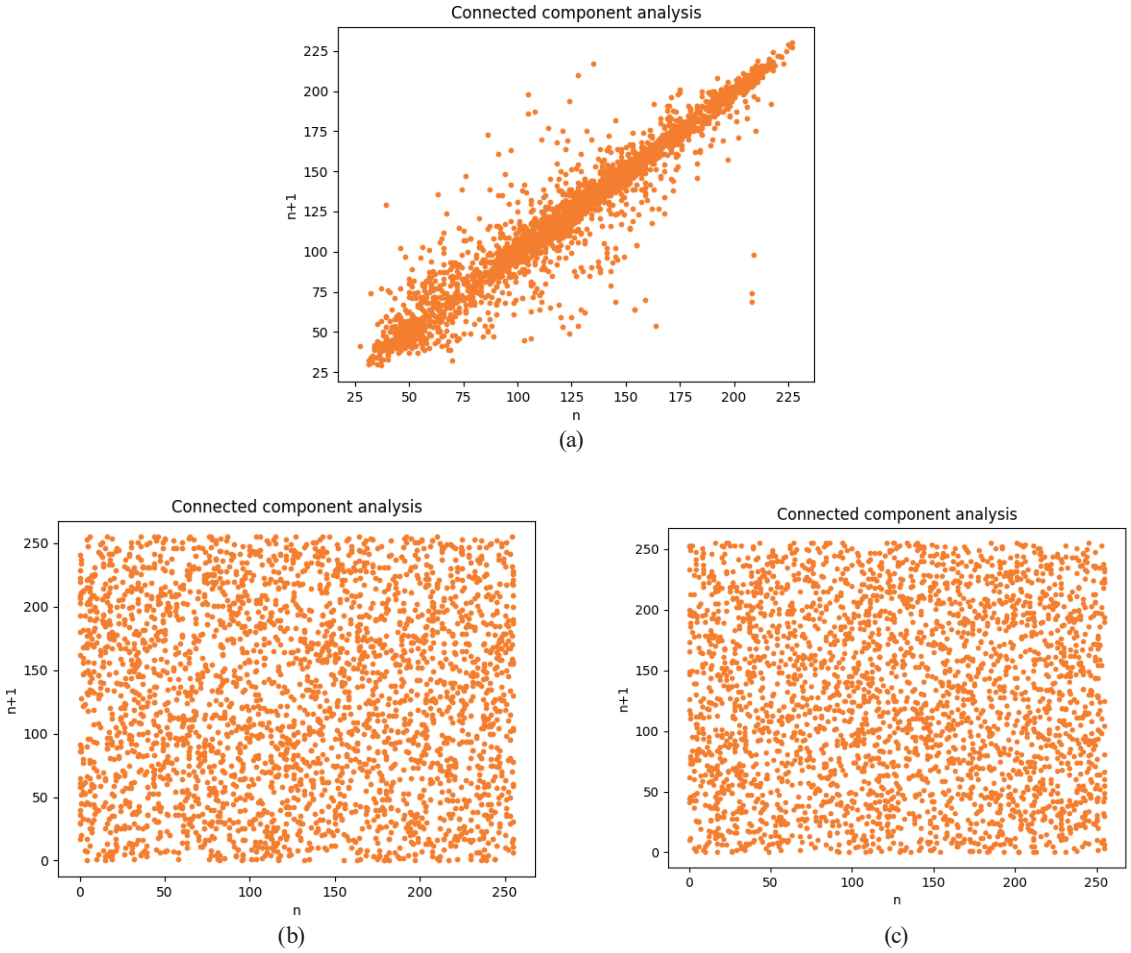


Fig. 11. (Color online) Connected component analysis: (a) original image, (b) cipher image with one key, and (c) cipher image with five keys.

cipher images generated with one key and five keys are distributed randomly and uniformly, as shown in Figs. 11(b) and 11(c), respectively. Table 3 provides detailed reports of the CCA. According to the results in Table 3, the cipher image produced with five dynamic keys exhibits the lowest pixel correlation, outperforming the results in previous reports.[25–28]

Information entropy (IE) analysis: Information entropy can represent a measure of event uncertainty. For encrypted images, the larger the value of the information entropy, the better the effect of the encryption. The calculation formula is introduced as

$$H = -\sum_{0}^{255} p_i(x) \log p_i(x), \tag{18}$$

where p_i denotes the frequency of each grayscale image. For a grayscale image, the pixel belongs to [0, 255] and the maximum value of IE will be 8. Figures 9(a)–9(c) are used for information entropy analysis. Table 4 shows the results of IE. According to Table 4, the cipher image by five keys obtains the largest IE, which is superior to the result in a previous report.[25]

Table 3
Connected component analysis with $N = 3000$.

Image	Plaint image	Cipher image with one key	Cipher image with five keys	Lin *et al.*[25] with five keys	Tang *et al.*[28]
Horizontal	0.9724	0.0076	2.763e−05	0.0061	−0.0685
Vertical	0.9852	0.0055	0.0013	0.0021	0.0857
Diagonal	0.9625	−0.0071	0.0021	−0.0045	0.0059
Absolute sum	2.9202	0.0203	0.0034	0.0127	0.1601

Table 4
Information entropy analysis.

	Original image	Cipher image by static key	Cipher image by five keys	Lin *et al.*[25] with five keys
Information entropy	6.9975653	7.991118	7.99153	7.9911

NPCR and UACI Analysis: NPCR and UACI are utilized to assess the sensitivity of an image in resisting differential attacks.[29] The formulas are defined as

$$D(i,j)=\begin{cases}0, if\ C_1(i,j)=C_2(i,j),\\ 1, if\ C_1(i,j)\neq C_2(i,j),\end{cases} \tag{19}$$

$$NPCR=\frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}D(i,j)\times 100\%, \tag{20}$$

$$UACI=\frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{\left|C_1(i,j)-C_2(i,j)\right|}{255}\times 100\%. \tag{21}$$

Here, M and N denote the width and height of the encrypted images, whereas $C_1(i, j)$ and $C_2(i, j)$ represent the pixel values at location (i, j) in the two ciphered images C_1 and C_2, respectively. For the NPCR and UACI tests, a pixel in Fig. 10(a) is randomly selected and adjusted by either increasing or decreasing its value by 1. Both the original image in Fig. 10(a) and the slightly altered image are encrypted using the five dynamic keys proposed in this paper. The NPCR test yields 99.6159%, whereas the UACI is 31.1153%. Furthermore, a comparison of the NPCR and UACI test outcomes between the proposed method and others is shown in Table 5. The comparison demonstrates that the proposed method outperforms previous methods[29–31] in both NPCR and UACI tests.

From the above analysis and comparisons, the proposed DCECRKs in this paper offer the following advantages and contributions:

(1) **Instantaneous synchronization design to improve system performance:** In this study, we introduced a discrete controller that enables the instantaneous synchronization of the master–slave chaotic system. It significantly reduces the transient time required for synchronization in existing chaotic synchronization research. This improvement enhances encryption and decryption system performance, making them more practical for real-world applications.

Table 5
NPCR and UACI tests.

Test methods	NPCR (%)	UACI (%)
This paper	99.6159	31.1153
Ferdush *et al.* [29]	99.5499	26.5199
Gupta *et al.* [30]	99.5899	28.4899
Bisht *et al.* [31]	99.7024	27.9796

(2) **Innovative DCECRK generation mechanism to enhance key randomness:** We proposed an innovative DCECRK generation mechanism, as shown in Fig. 5. Compared with existing chaotic encryption methods, the generated keys exhibit higher randomness and unpredictability. Additionally, by integrating synchronization techniques, synchronized DCECRKs are successfully designed. This ensures that both the transmitter and receiver generate identical dynamic keys, further improving encryption system reliability.
(3) **Integration of synchronized DCECRKs with the AES encryption algorithm to enhance image encryption security:** We combined synchronized DCECRKs with the AES encryption algorithm for image encryption and decryption. Test results demonstrate superior encryption performance, strengthening image data security. Furthermore, this method effectively addresses key management and distribution challenges in symmetric encryption algorithms. It ensures the safety of keys during transmission and storage, enhancing the overall reliability and robustness of the encryption system.

5. Conclusions

In this study, a chaotic extremum coding approach was newly developed to generate dynamic random numbers with good randomness quality. Through the proposed synchronization controllers, it is guaranteed that the transmitter and receiver can immediately and simultaneously generate the same dynamic chaotic extremum-coded random keys for encryption and decryption, respectively. Continuously, a cryptosystem with AES-CFB and dynamic synchronized random keys was proposed to guarantee the security of encrypted images. Finally, to highlight the contribution of the proposed extremum coding and cryptosystem, many tests including the statistical analysis, histogram, connected component analysis, information entropy, correlation indexes, number of pixels change rate, and unified average changing intensity were conducted, and the results were compared with those in the literature. The comparison results showed that the proposed cryptosystem provides better security for image encryption.

Declaration of Conflicting Interests

The author(s) declare no potential conflicts of interest to the research, authorship, and/or publication of this article.

Funding

This work was funded by the National Science and Technology Council of Taiwan under grant no. NSTC 113-2221-E-167-030.

ORCID iD

Jun-Juh Yan: https://orcid.org/0000-0002-9438-1937

References

1 I. Ahmad, T. Kumar, M. Liyanage, J. Okwuibe, M. Ylianttila, and A. Gurtov: 2017 IEEE Conf. Standards for Communications and Networking (CSCN) (2017) 193–199.
2 R. Yegireddi and R. K. Kumar: 2016 Int. Conf. ICT in Business Industry & Government (ICTBIG) (2016) 1.
3 P. W. Shor: SIAM Rev. **41** (1999) 303. https://doi.org/10.1137/S0036144598347011
4 L. K. Grover: Proc. 28th Annu. ACM Symp. Theory of Computing (1996) 212–219.
5 M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt: Post-Quantum Cryptography (Springer, Cham, 2016) pp. 29–43.
6 R. P. Brent: Int. Computing and Combinatorics Conf. (2000) 3–22.
7 F. Zhang and J. Wang: Ann. Phys. **451** (2023) 169263. https://doi.org/10.1016/j.aop.2023.169263
8 J. S. Fang, J. S. H. Tsai, J. J. Yan, L. H. Chiang, and S. M. Guo: Meas. Control **54** (2021) 1174. https://doi.org/10.1177/00202940211021116
9 Y. K. Zhang, Z. Y. Li, Z. Y. Tao, Y. Su, and Y. X. Fan: Opt. Commun. **528** (2023) 129017. https://doi.org/10.1016/j.optcom.2022.129017
10 H. A. Mansour: J. Inf. Hiding Multimedia Signal Process **12** (2021) 83.
11 J. Liu, Z. Wang, M. Shu, F. Zhang, S. Leng, and X. Sun: Complexity **2019** (2019). https://doi.org/10.1155/2019/7242791
12 K. M. Cuomo and A.V. Oppenheim: Phys. Rev. Lett. **71** (1993) 65. https://doi.org/10.1103/PhysRevLett.71.65
13 I. Koyuncu and A. T. Özcerit: Comput. Electr. Eng. **58** (2017) 203. https://doi.org/10.1016/j.compeleceng.2016.07.005
14 I. Koyuncu, M. Tuna, I. Pehlivan, C.B, Fidan, and M. Alçın: Analog Integr. Circuits Signal Process.**102** (2020) 445. https://doi.org/10.1007/s10470-019-01568-x
15 C. H. Lin, G. H. Hu, C. Y. Chan, and J.J. Yan: Appl. Sci. **11** (2021) 1329. https://doi.org/10.3390/app11031329
16 Q. Huang, L. Wang, and G. Li: 2018 10th Int. Conf. Measuring Technology and Mechatronics Automation (ICMTMA) (2018) 444–447.
17 S. Kassim, O. Megherbi, H. Hamiche, S. Djennoune, and M. Bettayeb: 2019 IEEE Int. Symp. Signal Processing and Information Technology (ISSPIT) (2019) 1–6.
18 M. J. Dworkin, E. B. Barker, J. R. Nechvatal, J. Foti, L. E. Bassham, E. Roback, and J. F. J. Dray: Announcing the advanced encryption standard (AES) NIST FIPS 197, 2001. https://www.nist.gov/publications/advanced-encryption-standard-aes
19 A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, (2010) A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST Special Publication 800-22.
20 K. D. Young, V. I. Utkin, and U. Ozguner: IEEE Trans. Control Syst. Technol. **7** (1999) 328. https://doi.org/10.1109/87.761053
21 J. Liu: Sensor Networks and Signal Processing (Singapore, 2021) 341–354. https://doi.org/10.1007/978-981-15-4917-5_26
22 P. Li, L. Min, Y. Hu, G. Zhao, and X. Li: 2012 IEEE 6th Int. Conf. Information and Automation for Sustainability (2012) 159–162.
23 A. Modiri and S. Mobayen: ISA Trans. **105** (2020) 33. https://doi.org/10.1016/j.isatra.2020.05.039
24 N. Charalampidis, C. Volos, L. Moysis, H.E. Nistazakis, and I. Stouboulos: 2022 11th Int. Conf. Modern Circuits and Systems Technologies (MOCAST) (2022) 1–4. https://doi: 10.1109/MOCAST54814.2022.9837545
25 C. H. Lin, G.H. Hu, J. S. Chen, J. J. Yan, and K. H. Tang: Multimedia Syst. **28** (2022) 1793. https://doi.org/10.1007/s00530-022-00950-6

26 A. Ozdemir, I. Pehlivan, A. Akgul, and E. Guleryuz: Chin. J. Phys. **56** (2018) 2852. https://doi.org/10.1016/j.cjph.2018.09.021
27 W. Zhang, K. W. Wong, H. Yu, and Z. L. Zhu: Commun. Nonlinear Sci. Numer. Simul. **18** (2013) 2066. https://doi.org/10.1016/j.cnsns.2012.12.012
28 Z. Tang, Y. Yang, S. Xu, C. Yu, and X. Zhang: Security Commun. Networks **15** (2019) 1. https://doi.org/10.1155/2019/8694678
29 J. Ferdush, M. Begum, and M. S. Uddin: Adv. Multimedia **2021** (2021) 1. https://doi.org/10.1155/2021/5527295
30 M. Gupta, K. K. Gupta, and P. K. Shukla: Multimed. Tools Appl. **80** (2020) 10391. https://doi.org/10.1007/s11042-020-10116-z
31 A. Bisht, M. Dua, S. Dua, and P. Jaroli: J. Intell. Syst. **342** (2019) 1246. https://doi.org/10.1515/jisys-2018-0365

About the Authors

Chih-Hsueh Lin received his Ph.D. degree in computer science and engineering from National Sun Yat-Sen University, Kaohsiung City, in 2006. Since 2024, he has been a professor with the Electronic Engineering Department, National Kaohsiung University of Science and Technology, Kaohsiung City, Taiwan. His research interests include machine learning, information security, biomedical signal processing, and data mining. (cslin@nkust.edu.tw)

Guo-Hsin Hu received his Ph.D. degree in electronic engineering from National Kaohsiung University of Science and Technology, Taiwan. His research interest focuses on smart manufacturing, text mining, and information technology. (guohsin@mail.mirdc.org.tw)

Shunsuke Araki received his B.E., M.E., and Dr. Eng. degrees in computer science and system engineering from Kyushu Institute of Technology, Japan in 1996, 1998, and 2003, respectively. From 2000, he was a research associate at Kyushu Institute of Technology, and he became an associate professor in 2018. His main research interests are pseudorandom number generators, digital signatures, and IoT secure protocols. (araki@csn.kyutech.ac.jp)

Hong-Lin Mu received his B.S. and M.S. degrees from the Department of Electronic Engineering, National Chin-Yi University of Technology, Taiwan in 2021 and 2023, respectively. His main research interests are in circuit implementation, chaotic systems, and nonlinear control. (hunglin0927@gmail.com)

Jun-Juh Yan received his M.S. degree in electrical engineering from National Central University, Taiwan in 1992 and his B.S. and Ph.D. degrees in electrical engineering from National Cheng Kung University, Taiwan, in 1987 and 1998, respectively. At present, he is a professor in the Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung, Taiwan. His main research interests are in multirobot dynamic systems, chaotic systems, neural networks, variable-structure control systems, and adaptive control. (jjyan@ncut.edu.tw)