PLOS ONE

RESEARCH ARTICLE

A semi-symmetric image encryption scheme based on the function projective synchronization of two hyperchaotic systems

Xiaoqiang Di*, Jinqing Li, Hui Qi, Ligang Cong, Huamin Yang

School of Computer Science and Technology, Changchun University of Science and Technology, Changchun, JiLin Province, China

* dixiaoqiang@cust.edu.cn

Citation: Di X, Li J, Qi H, Cong L, Yang H (2017) A semi-symmetric image encryption scheme based on the function projective synchronization of two hyperchaotic systems. PLoS ONE 12(9): e0184586. https://doi.org/10.1371/journal.pone.0184586

Editor: Yeng-Tseng Wang, Kaohsiung Medical University, TAIWAN

Received: April 26, 2017

Accepted: August 26, 2017

Published: September 14, 2017

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This research is partially supported by Industrial Innovation Project of Jilin Province (2016C087, http://jldrc.gov.cn) and Science and Technology Project of Jilin Province (20150312030ZX, http://www.jlkjt.gov.cn/bsfw/kjjhxmsb/). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Abstract

Both symmetric and asymmetric color image encryption have advantages and disadvantages. In order to combine their advantages and try to overcome their disadvantages, chaos synchronization is used to avoid the key transmission for the proposed semi-symmetric image encryption scheme. Our scheme is a hybrid chaotic encryption algorithm, and it consists of a scrambling stage and a diffusion stage. The control law and the update rule of function projective synchronization between the 3-cell quantum cellular neural networks (QCNN) response system and the 6^{th}-order cellular neural network (CNN) drive system are formulated. Since the function projective synchronization is used to synchronize the response system and drive system, Alice and Bob got the key by two different chaotic systems independently and avoid the key transmission by some extra security links, which prevents security key leakage during the transmission. Both numerical simulations and security analyses such as information entropy analysis, differential attack are conducted to verify the feasibility, security, and efficiency of the proposed scheme.

Introduction

With the rapid growth of broadband communication, multimedia transmission has increased over the Internet, which makes information and communication systems more vulnerable. Image security has attracted a huge amount of attention due to the widespread interconnection of almost all devices and communication networks. Image encryption differs from text encryption due to bulk data capacity, high redundancy and a strong correlation between adjacent pixels.

Since Matthews [1] first proposed the chaos encryption algorithm in 1989, many studies have indicated that chaotic encryption are suitable for bulk data due to its favorable properties, such as complex and nonlinear, high sensitivity to initial conditions, control parameters, non-periodicity, and a pseudorandom nature.

Many image encryption algorithms [2–25] based on chaos have been developed in last decades to ensure the security of digital images transmission and storage. Most of them

Competing interests: The authors have declared that no competing interests exist.

adopted permutation-diffusion mechanism [3–7, 11, 16, 17, 19, 21], in which permuting the positions of image pixels incorporates with changing gray values of image pixels to confuse the relationship between the cipher image and the plain image.

A previous study [26]proposed an image encryption/decryption algorithm with compound chaos mapping, in addition, a hyperchaotic system based on chaotic control parameters was put forward. Another study [27] presented an image encryption scheme on the foundation of multiple chaotic maps while an alternate work [28] proposed an image encryption algorithm on the basis of rotation matrix bit-level permutation and block diffusion. Akram Belazi proposed several image encryption schemes [23–25] based on chaos and obtained the good encryption effect. An encryption method on the basis of reversible cellular automata combined with chaos has also been designed in [12]. Ref [29] presented a color image encryption scheme on the foundation of the quantum chaotic system.

According to the type of the key usage, encryption algorithm can be divided into symmetric encryption and asymmetric encryption. The same secret key is used to encrypt and decrypt in symmetric encryption algorithms. Most chaos image encryption schemes are based on symmetric cryptographic techniques, which have been proven to be more vulnerable than an asymmetric cryptosystem [30].

Common symmetric encryption algorithms include DES, 3DES and AES. They are widely used due to their advantages such as great speed, relatively low complexity as well as easy implementation in hardware. Since both encryption and decryption sides should configure the key by some extra methods, once the key is divulged the cryptosystems will be broken. Furthermore, each pair of users need choose a unique key that nobody else knows. This makes the quantity of key to be growing exponentially.

Asymmetric encryption differs from symmetric encryption that it requires a key pair: a public key for encryption and a corresponding private key for decryption which is known only to the owner. The most common asymmetric encryption algorithm is RSA. In an asymmetric key cryptosystem, any user can encrypt a message using the public key of the receiver, but such a message can be decrypted only with the receiver's private key [31]. It is unlike symmetric encryption to share the key, asymmetric encryption do not require a secure channel for the initial exchange of the key from transmitter to receiver. Although asymmetric cryptosystem has so many advantages, it also has disadvantages. For example, it is extremely difficult to factorize large numbers in order to obtain sufficiently long keys especially enormous data.

Since Pecora and Corrall found the drive-response chaos synchronization phenomena [32], a lot of synchronization schemes have been proposed, such as complete synchronization, generalized synchronization, phase synchronization, lag synchronization, projective synchronization. Two chaotic systems synchronization phenomenon is similar to the asymmetric key mechanism, and they can synchronize with each other if they exchange information in just the right way. This motivates us to use chaos synchronization to avoid the key transmission in order to combine the advantage of the symmetric and asymmetric encryption and try to overcome their shortcomings.

In this paper, we propose a new color image cryptosystem using a synchronization scheme for a 3-cell QCNN [33] and a 6^{th}-order CNN [34]. The 3-cell QCNN is regarded as the response system and the 6th order CNN is used for the drive system. In order to synchronize the drive-response system, the control law for stable synchronization errors and the update rule for unknown parameters estimation are given. The function projective synchronization [35] is treated as the decryption key generator. We prove that the 6^{th}-order CNN drive system and the 3-cell QCNN response system are asymptotically synchronized. Numerical simulations and security analyses such as information entropy analysis, differential attack are performed to verify the feasibility of the proposed scheme. As similar as the asymmetric

encryption, our scheme does not require exchange key, and it effectively avoids the threat of key exposure, therefore, it will be called Semi-Symmetric encryption scheme.

The rest of the paper is organized as follows: In the next section, we briefly describe the 3-cell QCNN system and the 6^{th}-order CNN system used in our scheme. In Section 3, the function projection synchronization between the response system and the drive system is presented. Section 4 gives the semi-symmetric encryption scheme. The experimental results and performance analyses are given in Section 5. Section 6 concludes the paper.

System descriptions

3-cell QCNN hyperchaotic system

Quantum dots and quantum cellular automata (QCA) [36] constitute new types of semiconductor nano-materials that have many unique nano-features. The k^{th} QCA state equation is obtained by the Schrödinger equation [36]:

$$\begin{aligned} i\hbar\frac{\partial}{\partial t}P_k &= -2\gamma\sqrt{1-P_k^2}\sin\varphi_k \\ i\hbar\frac{\partial}{\partial t}\varphi_k &= -E_k\bar{P}_k + 2\gamma\frac{P_k}{\sqrt{1-P_k^2}}\cos\varphi_k \end{aligned} \tag{1}$$

where $\hbar$ is Planck's constant, γ is the inter-dot tunneling energy, which takes into account the neighboring polarizations, and E_k is the electrostatic energy cost of two adjacent fully polarized cells that have opposite polarization. The effect of local interconnections is considered in the term $\bar{P}_k$; and φ_k is a quantum phase of the QCA. Eq (1) constitutes the QCNN state equations and its dynamics are characterized by two variables, P_k and ϕ_k. A 3–cell QCNN system can be described as Eq (2):

$$\begin{aligned} \dot{P}_1 &= -2b_{01}\sqrt{1-P_1^2}\sin\phi_1 \\ \dot{\phi}_1 &= -\omega_{01}(P_1-P_2-P_3) + 2b_{01}\frac{P_1}{\sqrt{1-P_1^2}}\cos\phi_1 \\ \dot{P}_2 &= -2b_{02}\sqrt{1-P_2^2}\sin\phi_2 \\ \dot{\phi}_2 &= -\omega_{02}(P_2-P_1-P_3) + 2b_{02}\frac{P_2}{\sqrt{1-P_2^2}}\cos\phi_2 \\ \dot{P}_3 &= -2b_{03}\sqrt{1-P_3^2}\sin\phi_3 \\ \dot{\phi}_3 &= -\omega_{03}(P_3-P_1-P_2) + 2b_{03}\frac{P_3}{\sqrt{1-P_3^2}}\cos\phi_3 \end{aligned} \tag{2}$$

where P_1, P_2, P_3 and ϕ_1, ϕ_2, ϕ_3 are the state variables; b_{01}, b_{02}, and b_{03} are the proportional inter–dot energy in each cell, and ω_{01}, ω_{02}, ω_{03} are effect weigh parameters on the differences in the polarization of the adjacent cells, like the cloning templates in traditional CNNs. The Fig 1, shows the attractor of system(2) in three dimensional space. We investigated the dynamic behavior of system(2) by calculating its Lyapunov exponents. When $b_{01} = b_{02} = b_{03} = 0.28$, $\omega_{01} = 0.5$, $\omega_{02} = 0.2$, and $\omega_{03} \in [0, 1]$, which are shown in Fig 2, system(2) is hyperchaotic due to three positive Lyapunov exponents.

6^{th}-order CNN hyperchaotic system

The 6^{th}-order CNN is another hyperchaotic system used in this paper, which is introduced in Ref [34], and it is all the interconnection in a CNN. Its state equation is defined as Eq (3):

$$\frac{dx_i}{dt} = -x_j + a_j p_j + \sum_{\substack{k=1 \\ k\neq j}}^{6} a_{j,k}p_k + \sum_{k=1}^{6} s_{jk}x_k + i_j \qquad (j = 1, 2, ..., 6) \tag{3}$$

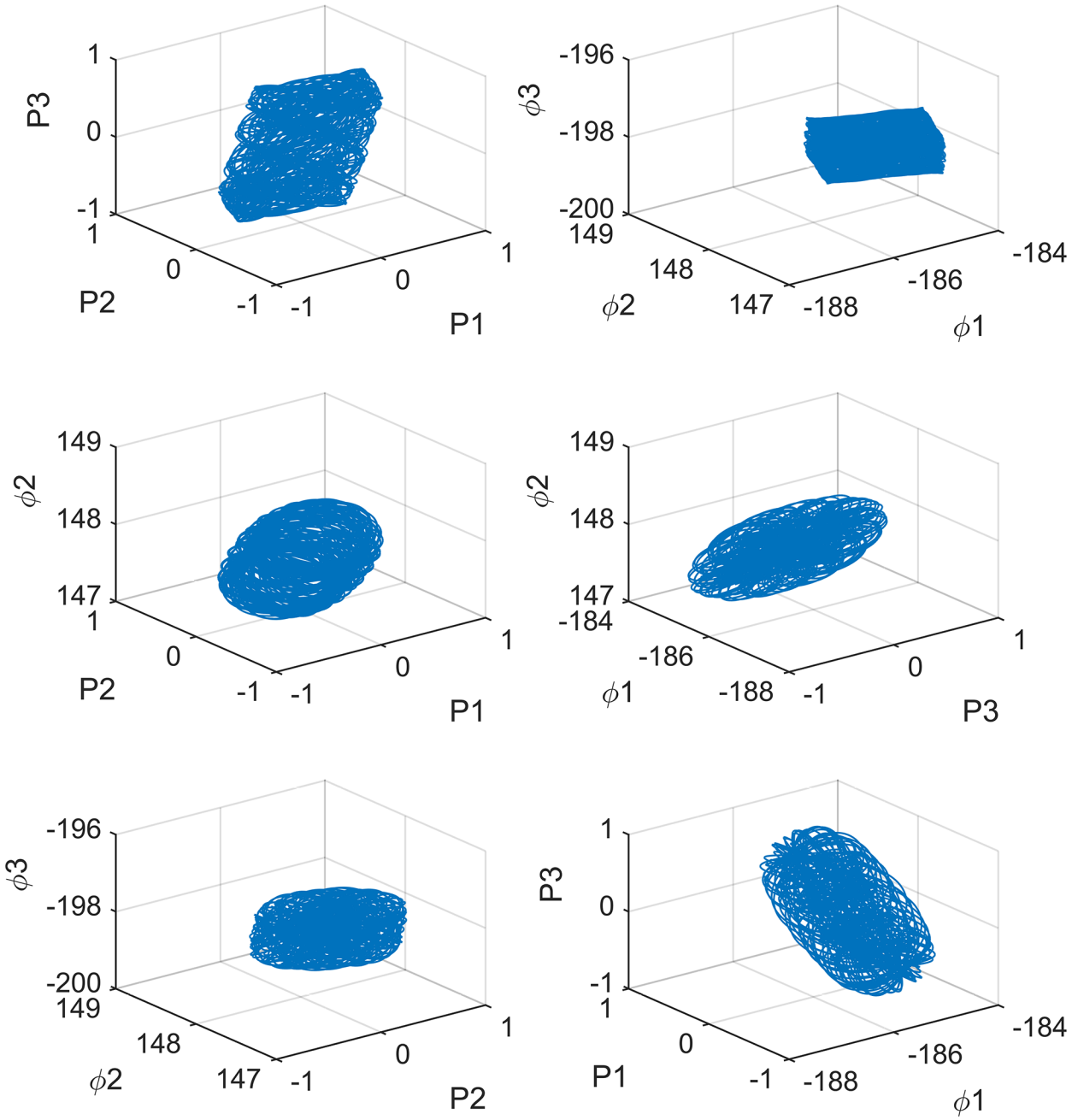

Fig 1. 3-cell QCNN system partial attractor distribution.

https://doi.org/10.1371/journal.pone.0184586.g001

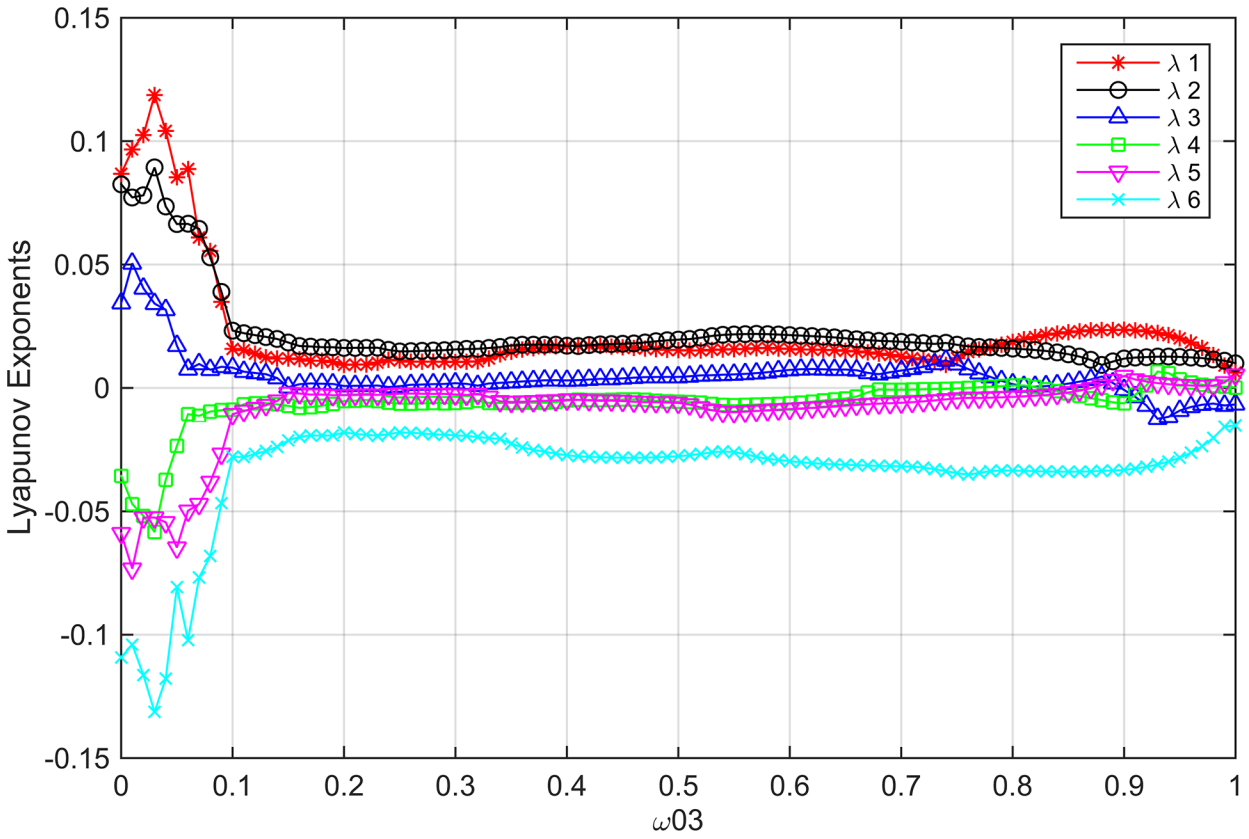

Fig 2. 3-cell QCNN system Lyapunov exponents spectrum with $b_{01} = b_{02} = b_{03} = 0.28$, $\omega_{01} = 0.5$, $\omega_{02} = 0.2$, and $\omega_{03} \in [0, 1]$.

https://doi.org/10.1371/journal.pone.0184586.g002

where

$$
\begin{aligned}
&a_j = 0 (j = 1, 2, 3, 5, 6), a_4 = 200;\\
&a_{jk} = 0 (j, k = 1, 2, ..., 6; j \neq k);\\
&s_{12} = s_{21} = s_{24} = s_{34} = s_{42} = s_{43} = s_{53} = s_{54} = s_{55} = s_{56} = s_{61} = s_{63} = s_{64} = 0;\\
&i_j = 0 (j = 1, 2, ..., 6);\\
&s_{11} = s_{23} = s_{33} = s_{51} = 1; s_{13} = s_{14} = -1;\\
&s_{22} = 3, s_{31} = 14, s_{32} = -14, s_{41} = s_{62} = 100, s_{44} = -99, s_{52} = 18, s_{65} = 4, s_{66} = -3;
\end{aligned}
$$

Eq (3) could be calculted as Eq (4):

$$
\begin{aligned}
\dot{x}_1 &= -x_3 - x_4\\
\dot{x}_2 &= 2x_2 + x_3\\
\dot{x}_3 &= 14x_1 - 14x_2\\
\dot{x}_4 &= 100x_1 - 100x_4 + 200p_4\\
\dot{x}_5 &= 18x_2 + x_1 - x_5\\
\dot{x}_6 &= 4x_5 - 4x_6 + 100x_2
\end{aligned}
\tag{4}
$$

where $p_4 = 0.5(|x_4 + 1| - |x_4 - 1|)$.

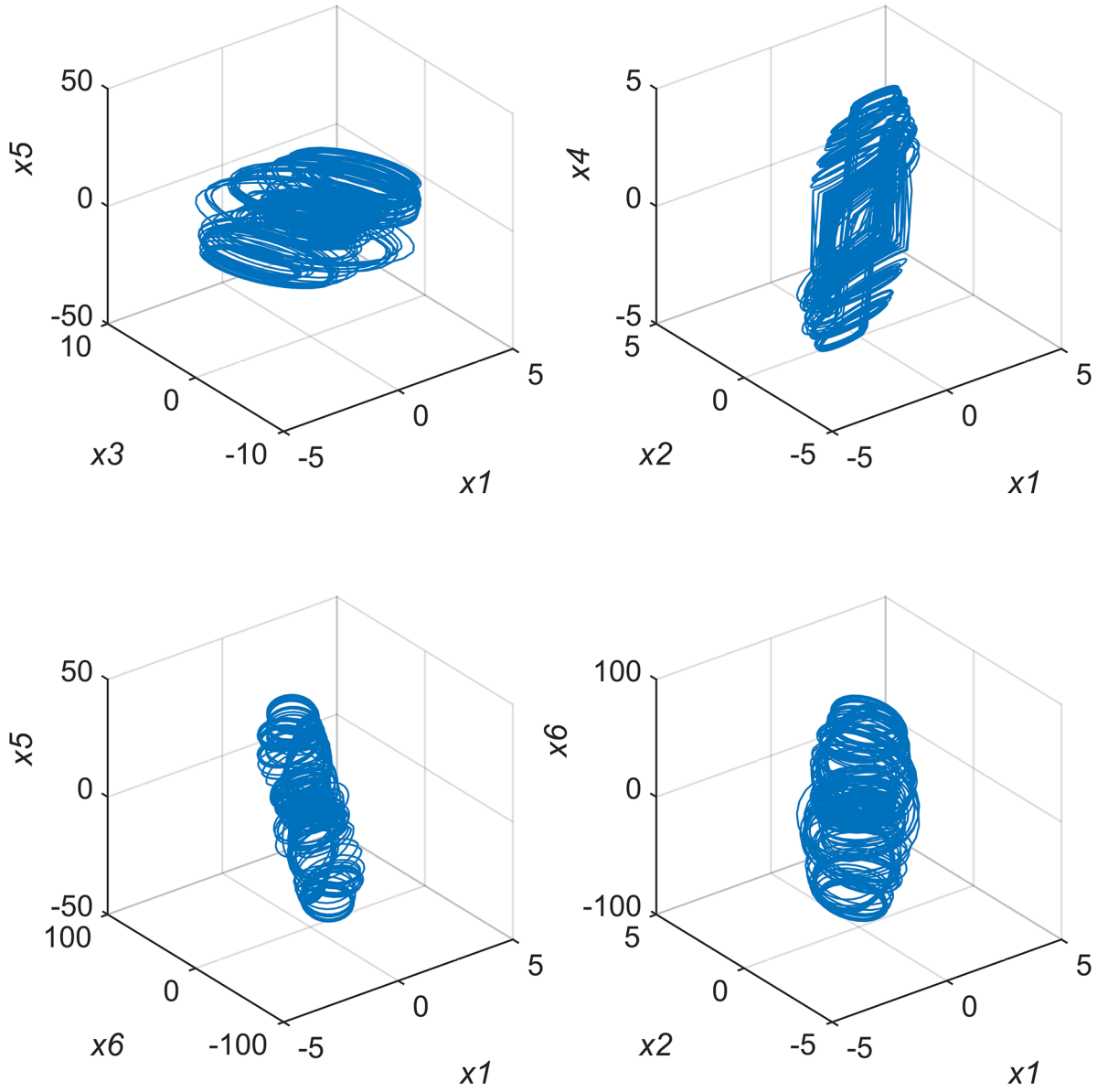

Fig 3. 6^{th}-order CNN partial chaotic attractor distribution.

https://doi.org/10.1371/journal.pone.0184586.g003

We calculated the Lyapunov exponents of system(4). When $t \to \infty$, the six Lyapunov exponents are $\lambda_1 = 2.748$, $\lambda_2 = -2.9844$, $\lambda_3 = 1.2411$, $\lambda_4 = -14.4549$, $\lambda_5 = -1.4123$ and $\lambda_6 = -83.2282$. Two of these exponents are positive, so system(4) is also hyperchaotic. Fig 3 shows system(4)'s partial chaotic attractor distribution.

The synchronized key generation system

Let System(4) and System(2) be the drive system and the response system, respectively. Thus, the system(2) can be described by the Eq (5) via the function projective

synchronization [35]:

$$\begin{aligned}
\dot{P}_{r1} &= -2b_{11}\sqrt{1-P_{r1}^2}\sin\phi_{r1}+u_1\\
\dot{\phi}_{r1} &= -\omega_{11}(P_{r1}-P_{r2}-P_{r3})+2b_{11}\frac{P_{r1}}{\sqrt{1-P_{r1}^2}}\cos\phi_{r1}+u_2\\
\dot{P}_{r2} &= -2b_{12}\sqrt{1-P_{r2}^2}\sin\phi_{r2}+u_3\\
\dot{\phi}_{r2} &= -\omega_{12}(P_{r2}-P_{r1}-P_{r3})+2b_{12}\frac{P_{r2}}{\sqrt{1-P_{r2}^2}}\cos\phi_{r2}+u_4\\
\dot{P}_{r3} &= -2b_{13}\sqrt{1-P_{r3}^2}\sin\phi_{r3}+u_5\\
\dot{\phi}_{r3} &= -\omega_{13}(P_{r3}-P_{r1}-P_{r2})+2b_{13}\frac{P_{r3}}{\sqrt{1-P_{r3}^2}}\cos\phi_{r3}+u_6
\end{aligned} \tag{5}$$

where b_{11}, b_{12}, b_{13}, ω_{11}, ω_{12} and ω_{13} are the parameters of response system(5) that need to be estimated in order to synchronize system(4) and system(5), and u_1, u_2, u_3, u_4, u_5 and u_6 are the controllers. Define synchronization error states as follows:

$$\dot{e}_i = \dot{y}_i - \alpha(t)\dot{x}_i - \dot{\alpha}(t)x_i, i = 1,2,3,4,5,6 \tag{6}$$

which $\dot{e}_i$ denotes the deviation between system(4) and system(5), when $\dot{e}_i$ converges to zero as time tends to infinity $\lim_{t\to\infty}||e_i|| = \lim_{t\to\infty}||y_i - \alpha(t)x_i|| = 0, i = 1,2,3,4,5,6$, $\alpha(t)$ as the scaling function factor, drive system and response system reach synchronization. Substituting Eqs (2), (4) and (5) into Eq (6) yields the error dynamical system(7) as defined in Eq (7) between system(4) and system(5):

$$\begin{aligned}
\dot{e}_1 &= -2b_{11}\sqrt{1-P_{r1}^2}\sin\varphi_{r1}+u_1-\alpha(t)(-2b_{01}\sqrt{1-P_1^2}\sin\varphi_1)-\dot{\alpha}(t)P_1\\
\dot{e}_2 &= -\omega_{11}(P_{r1}-P_{r2}-P_{r3})+2b_{11}\frac{P_{r1}}{\sqrt{1-P_{r1}^2}}\cos\varphi_{r1}+u_2\\
&\quad -\alpha(t)\left[-\omega_{01}(P_1-P_2-P_3)+2b_{01}\frac{P_1}{\sqrt{1-P_1^2}}\cos\varphi_1\right]-\dot{\alpha}(t)\varphi_1\\
\dot{e}_3 &= -2b_{12}\sqrt{1-P_{r2}^2}\sin\varphi_{r2}+u_3-\alpha(t)(-2b_{02}\sqrt{1-P_2^2}\sin\varphi_2)-\dot{\alpha}(t)P_2\\
\dot{e}_4 &= -\omega_{12}(P_{r2}-P_{r1}-P_{r3})+2b_{12}\frac{P_{r2}}{\sqrt{1-P_{r2}^2}}\cos\varphi_{r2}+u_4\\
&\quad -\alpha(t)\left[-\omega_{02}(P_2-P_1-P_3)+2b_{02}\frac{P_2}{\sqrt{1-P_2^2}}\cos\varphi_2\right]-\dot{\alpha}(t)\varphi_2\\
\dot{e}_5 &= -2b_{13}\sqrt{1-P_{r3}^2}\sin\varphi_{r3}+u_5-\alpha(t)(-2b_{03}\sqrt{1-P_3^2}\sin\varphi_3)-\dot{\alpha}(t)P_3\\
\dot{e}_6 &= -\omega_{13}(P_{r3}-P_{r1}-P_{r2})+2b_{13}\frac{P_{r3}}{\sqrt{1-P_{r3}^2}}\cos\varphi_{r3}+u_6\\
&\quad -\alpha(t)\left[-\omega_{03}(P_3-P_1-P_2)+2b_{03}\frac{P_3}{\sqrt{1-P_3^2}}\cos\varphi_3\right]-\dot{\alpha}(t)\varphi_3
\end{aligned} \tag{7}$$

We design the control law $u_i(i = 1, 2, 3, 4, 5, 6)$ as Eq (8) to make the synchronization errors e_1, e_2, e_3, e_4, e_5, and e_6 to stabilize at the origin.

$$\begin{aligned}
u_1 &= 2b_{11}[\sqrt{1-P_{r1}^2}\sin\varphi_{r1} - \alpha(t)\sqrt{1-P_1^2}\sin\varphi_1] + \dot{\alpha}(t)P_1 - k_1e_1 \\
u_2 &= \omega_{11}[(P_{r1} - P_{r2} - P_{r3}) - \alpha(t)(P_1 - P_2 - P_3)] \\
&\quad -2b_{11}\left[\frac{P_{r1}}{\sqrt{1-P_{r1}^2}}\cos\varphi_{r1} - \alpha(t)\frac{P_1}{\sqrt{1-P_1^2}}\cos\varphi_1\right] + \dot{\alpha}(t)\varphi_1 - k_2e_2 \\
u_3 &= 2b_{12}[\sqrt{1-P_{r2}^2}\sin\varphi_{r2} - \alpha(t)\sqrt{1-P_2^2}\sin\varphi_2] + \dot{\alpha}(t)P_2 - k_3e_3 \\
u_4 &= \omega_{12}[(P_{r2} - P_{r1} - P_{r3}) - \alpha(t)(P_2 - P_1 - P_3)] \\
&\quad -2b_{12}\left[\frac{P_{r2}}{\sqrt{1-P_{r2}^2}}\cos\varphi_{r2} - \alpha(t)\frac{P_2}{\sqrt{1-P_2^2}}\cos\varphi_2\right] + \dot{\alpha}(t)\varphi_2 - k_4e_4 \\
u_5 &= 2b_{13}[\sqrt{1-P_{r3}^2}\sin\varphi_{r3} - \alpha(t)\sqrt{1-P_3^2}\sin\varphi_3] + \dot{\alpha}(t)P_3 - k_5e_5 \\
u_6 &= \omega_{13}[(P_{r3} - P_{r1} - P_{r2}) - \alpha(t)(P_3 - P_1 - P_2)] \\
&\quad -2b_{13}\left[\frac{P_{r3}}{\sqrt{1-P_{r3}^2}}\cos\varphi_{r3} - \alpha(t)\frac{P_3}{\sqrt{1-P_3^2}}\cos\varphi_3\right] + \dot{\alpha}(t)\varphi_3 - k_6e_6
\end{aligned} \tag{8}$$

Furthermore, the update rule for the six unknown parameters b_{11}, b_{12}, b_{13}, ω_{11}, ω_{12}, and ω_{13} are Eq (9) defined as follows:

$$\begin{aligned}
\dot{b}_{11} &= 2\alpha(t)\sqrt{1-P_1^2}\sin\phi_1e_1 - 2\alpha(t)\frac{P_1}{\sqrt{1-P_1^2}}\cos\phi_1e_2 - k_7e_a \\
\dot{\omega}_{11} &= \alpha(t)(P_1 - P_2 - P_3)e_2 - k_8e_b \\
\dot{b}_{12} &= 2\alpha(t)\sqrt{1-P_2^2}\sin\phi_2e_3 - 2\alpha(t)\frac{P_2}{\sqrt{1-P_2^2}}\cos\phi_2e_4 - k_9e_c \\
\dot{\omega}_{12} &= \alpha(t)(P_2 - P_1 - P_3)e_4 - k_{10}e_d \\
\dot{b}_{13} &= 2\alpha(t)\sqrt{1-P_3^2}\sin\phi_3e_5 - 2\alpha(t)\frac{P_3}{\sqrt{1-P_3^2}}\cos\phi_3e_6 - k_{11}e_e \\
\dot{\omega}_{13} &= \alpha(t)(P_3 - P_1 - P_2)e_6 - k_{12}e_f
\end{aligned} \tag{9}$$

Where $k_i > 0(i = 1, 2, 3, \ldots, 12)$, and $e_a = b_{11}-b_{01}$, $e_b = \omega_{11} - \omega_{01}$, $e_c = b_{12} - b_{02}$, $e_d = \omega_{12} - \omega_{02}$, $e_e = b_{13} - b_{03}$, $e_f = \omega_{13} - \omega_{03}$.

Theorem. For a given nonzero scaling function factor $\alpha(t)$, it can make response system(5) and drive system(4) to synchronize by the control law Eq (8) and the update rule Eq (9).

Proof. Choose the following Lyapunov function:

$$V = \frac{1}{2}(e_1^2 + e_2^2 + e_3^2 + e_4^2 + e_5^2 + e_6^2 + e_a^2 + e_b^2 + e_c^2 + e_d^2 + e_e^2 + e_f^2)$$

The time derivative of V along the trajectory of the error system(6) is

$$\dot{V} = (e_1\dot{e}_1 + e_2\dot{e}_2 + e_3\dot{e}_3 + e_4\dot{e}_4 + e_5\dot{e}_5 + e_6\dot{e}_6 + e_a\dot{e}_a + e_b\dot{e}_b + e_c\dot{e}_c + e_d\dot{e}_d + e_e\dot{e}_e + e_f\dot{e}_f),$$

$$\begin{aligned}
\dot{V} &= e_1[-2(b_{11}-b_{01})\alpha(t)\sqrt{1-P_1^2}\sin\phi_1 - k_1e_1] \\
&\quad + e_2\left[-(\omega_{11}-\omega_{01})\alpha(t)(P_1-P_2-P_3) + 2(b_{11}-b_{01})\alpha(t)\frac{P_1}{\sqrt{1-P_1^2}}\cos\phi_1 - k_2e_2\right] \\
&\quad + e_3[-2(b_{12}-b_{02})\alpha(t)\sqrt{1-P_2^2}\sin\phi_2 - k_3e_3] \\
&\quad + e_4\left[-(\omega_{12}-\omega_{02})\alpha(t)(P_2-P_1-P_3) + 2(b_{12}-b_{02})\alpha(t)\frac{P_2}{\sqrt{1-P_2^2}}\cos\phi_2 - k_4e_4\right] \\
&\quad + e_5[-2(b_{13}-b_{03})\alpha(t)\sqrt{1-P_3^2}\sin\phi_3 - k_5e_5] \\
&\quad + e_6\left[-(\omega_{13}-\omega_{03})\alpha(t)(P_3-P_1-P_2) + 2(b_{13}-b_{03})\alpha(t)\frac{P_3}{\sqrt{1-P_3^2}}\cos\phi_3 - k_6e_6\right] \\
&\quad + e_a\left[2\alpha(t)\sqrt{1-P_1^2}\sin\phi_1e_1 - 2\alpha(t)\frac{P_1}{\sqrt{1-P_1^2}}\cos\phi_1e_2 - k_7e_a\right] \\
&\quad + e_b[\alpha(t)(P_1-P_2-P_3)e_2 - k_8e_b] \\
&\quad + e_c\left[2\alpha(t)\sqrt{1-P_2^2}\sin\phi_2e_3 - 2\alpha(t)\frac{P_2}{\sqrt{1-P_2^2}}\cos\phi_2e_4 - k_9e_c\right] \\
&\quad + e_d[\alpha(t)(P_2-P_1-P_3)e_4 - k_{10}e_d] \\
&\quad + e_e\left[2\alpha(t)\sqrt{1-P_3^2}\sin\phi_3e_5 - 2\alpha(t)\frac{P_3}{\sqrt{1-P_3^2}}\cos\phi_3e_6 - k_{11}e_e\right] \\
&\quad + e_f[\alpha(t)(P_3-P_1-P_2)e_6 - k_{12}e_f] \\
&= -k_1e_1^2 - k_2e_1^2 - k_3e_3^2 - k_4e_4^2 - k_5e_5^2 - k_6e_6^2 - k_7e_a^2 - k_8e_b^2 - k_9e_c^2 - k_{10}e_d^2 \\
&\quad - k_{11}e_e^2 - k_{12}e_f^2 \\
&= -e^TKe
\end{aligned}$$

where $e = (e_1, e_2, e_3, e_4, e_5, e_6, e_a, e_b, e_c, e_d, e_e, e_f)^T$, and $K = diag(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}, k_{11}, k_{12})^T$

Because $\dot{V} \leq 0$, we have $e_1, e_2, e_3, e_4, e_5, e_6, e_a, e_b, e_c, e_d, e_e, e_f \rightarrow 0$ as $t \rightarrow \infty$.
i.e., $\lim_{t\rightarrow\infty}||e|| = 0$. Proof completed.

Simulation is performed in order to evaluate the feasibly and effectiveness of the proposed control law and the update rule for the 3-cell QCNN and 6^{th}-order CNN synchronization method. The initial values and control parameters of the drive and the response system for a time-step of 0.1 are shown in Table 1.

In addition, the scaling function $\alpha(t) = 0.5 + 0.1\sin(t)$ and the control gains are defined as $(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}, k_{11}, k_{12}) = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$.

The simulations results are illustrated in Figs 4 and 5. Fig 4 shows that the errors e_1, e_2, e_3, e_4, e_5, and e_6 approach zero. Fig 5 shows that the estimated unknown parameters converge to $b_{11} \rightarrow 0.28$, $\omega_{11} \rightarrow 0.4$, $b_{12} \rightarrow 0.28$, $\omega_{12} \rightarrow 0.35$, $b_{13} \rightarrow 0.28$, and $\omega_{13} \rightarrow 0.25$ as $t \rightarrow \infty$. When t = 10, synchronization errors close 0 and unknown control parameters reach stability, which shows that the synchronization method is efficient.

Table 1. Initial values and control parameters of drive and response system.

Drive system	Response system	Control parameters of Response system
$x_1(0) = -0.92$	$P_{r1}(0) = 0.1901$	$b_{11} = 0.5$
$x_2(0) = 1.41$	$\phi_{r1}(0) = -184.3$	$\omega_{11} = 0.6$
$x_3(0) = -1.53$	$P_{r2}(0) = 0.123$	$b_{12} = 0.4$
$x_4(0) = 0.48$	$\phi_{r2}(0) = -147.3$	$\omega_{12} = 0.7$
$x_5(0) = 0.37$	$P_{r3}(0) = 0.113$	$b_{13} = 0.7$
$x_6(0) = -1.21$	$\phi_{r3}(0) = -197.85$	$\omega_{13} = 0.5$

https://doi.org/10.1371/journal.pone.0184586.t001

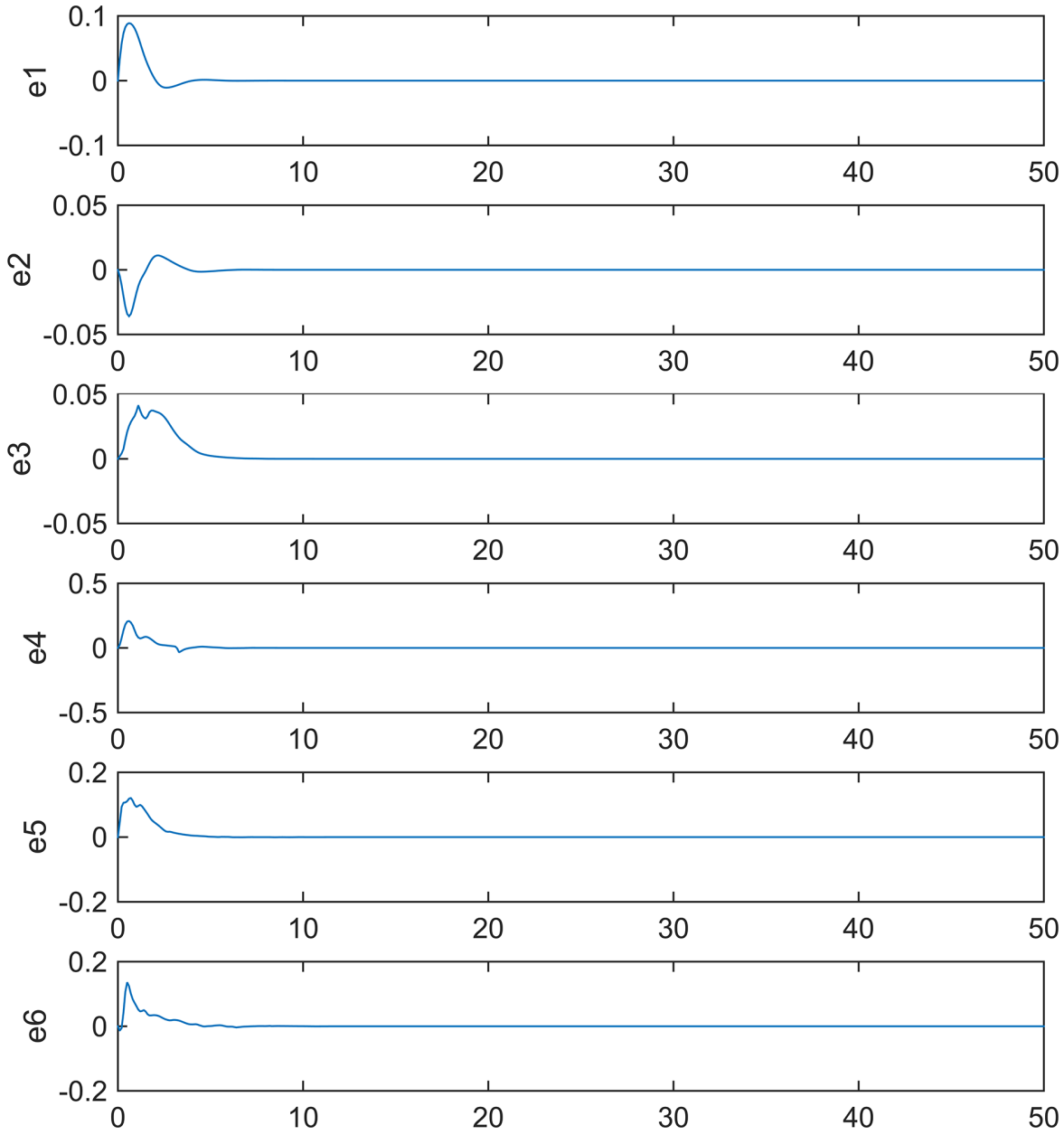

Fig 4. Error signals between the drive and the response system.

https://doi.org/10.1371/journal.pone.0184586.g004

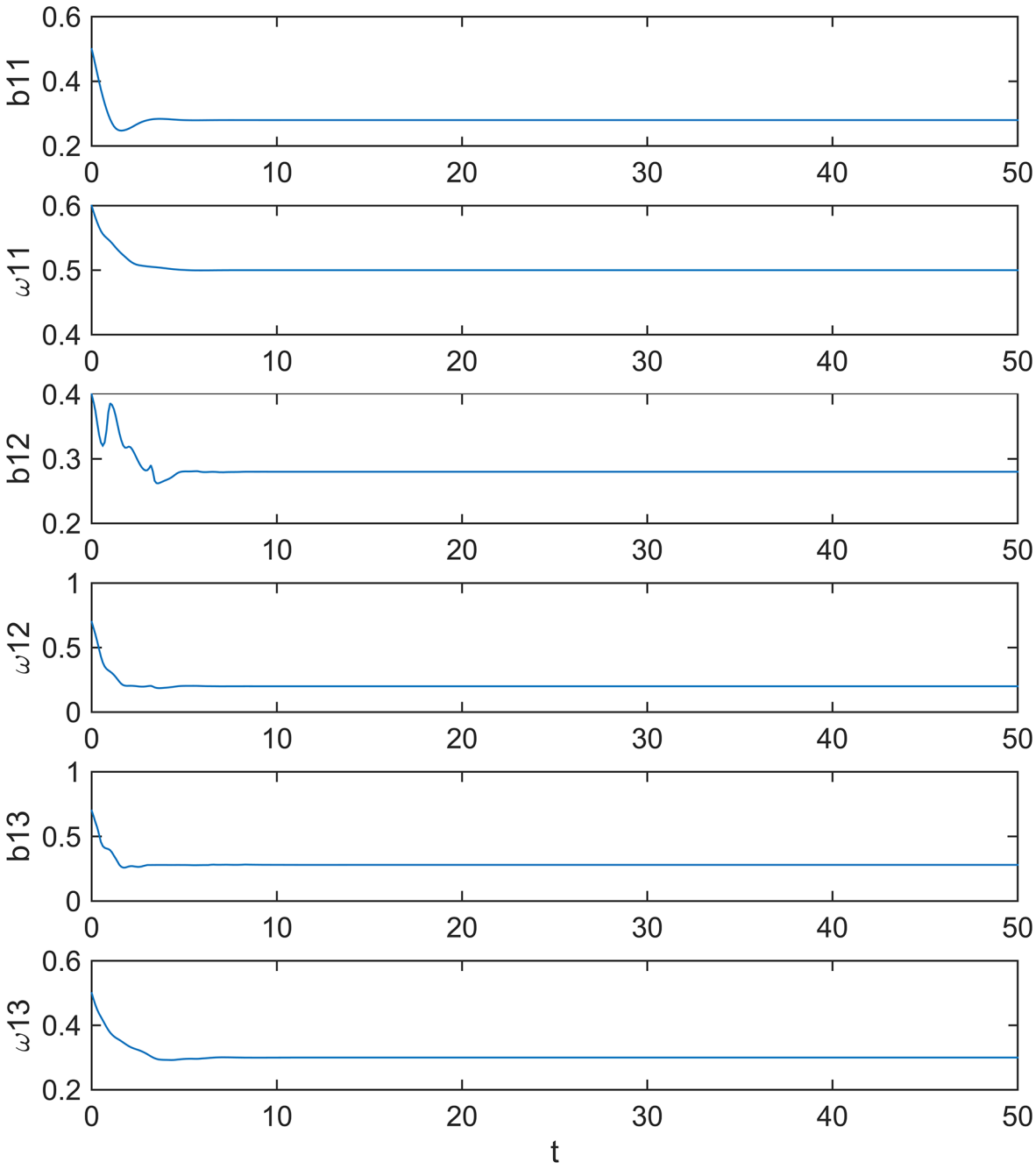

Fig 5. Estimated values for unknown parameters.

https://doi.org/10.1371/journal.pone.0184586.g005

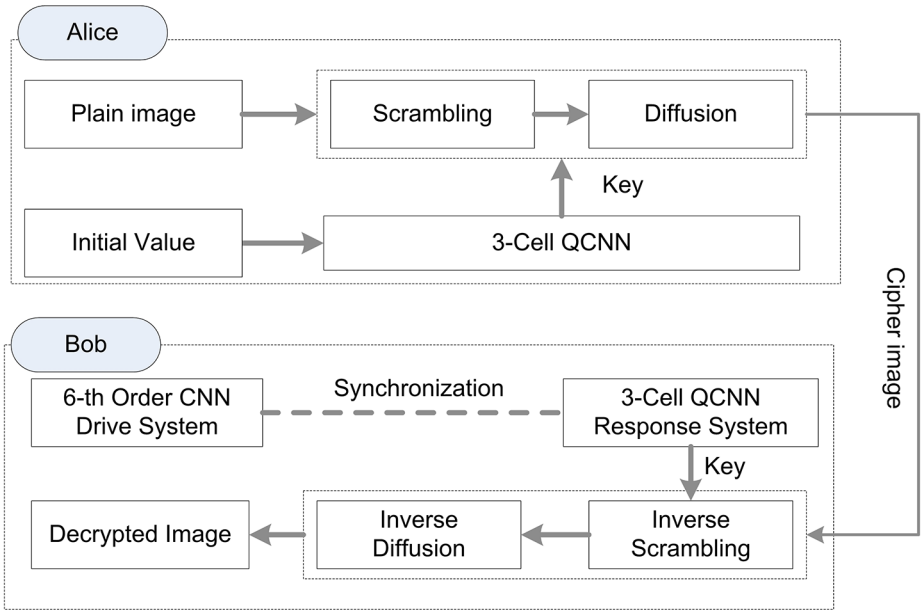

Fig 6. Semi-symmetric image encryption/decryption scheme.

https://doi.org/10.1371/journal.pone.0184586.g006

The semi-symmetric image encryption scheme

In this paper, we propose a semi-symmetric image encryption/decryption scheme based on the function projective synchronization. The proposed scheme is illustrated in Fig 6. The scheme is deployed at the ends of Alice and Bob, respectively. Firstly, Alice adopts system(2) with initial parameters and control parameters to obtain the key. Bob adopts system(5) to obtain the key independently. Function projective synchronization ensures that Alice and Bob get the equivalent key. Secondly, Alice encrypts the plain image by his key and transmits the cipher image to Bob. Thirdly, Bob decrypts the cipher image with his key.

The proposed scheme is different with symmetric algorithms that Alice and Bob use in different key generation systems. The symmetric algorithms transmit the key by some extra security methods. The proposed scheme is similar to asymmetric algorithms that the keys generated by the two systems need not transmit to each other over other security link, which prevents security key leakage during the transmission.

Our scheme is a hybrid chaotic encryption algorithm. It consists of a scrambling stage and a diffusion stage. In encryption phase, 3-cell QCNN system(2) is used for scrambling and diffusing the plain image. In decryption phase, since the function projective synchronization is used to synchronize the response system(5) and drive system(4), the 6^{th}-order CNN drive system(4) with control laws(8) and update rules(9) generates the key to decrypt the cipher image.

Encryption algorithm

This encryption flowchart is presented in Fig 7.

The 3-cell QCNN system(2) is the encryption key generator. The initial conditions $\phi_1(0)$, $\phi_2(0)$, $\phi_3(0)$, $P_1(0)$, $P_2(0)$, and $P_3(0)$ and control parameters b_{01}, b_{02}, b_{03}, ω_{01}, ω_{02}, and ω_{03} are used to iterate system(2) M times. The results are ϕ_1, ϕ_1, ϕ_3, P_1, P_2, and P_3 encryption keys. In the scrambling stage, the Arnold mapping [37] defined that Eq (10) is used to scramble the

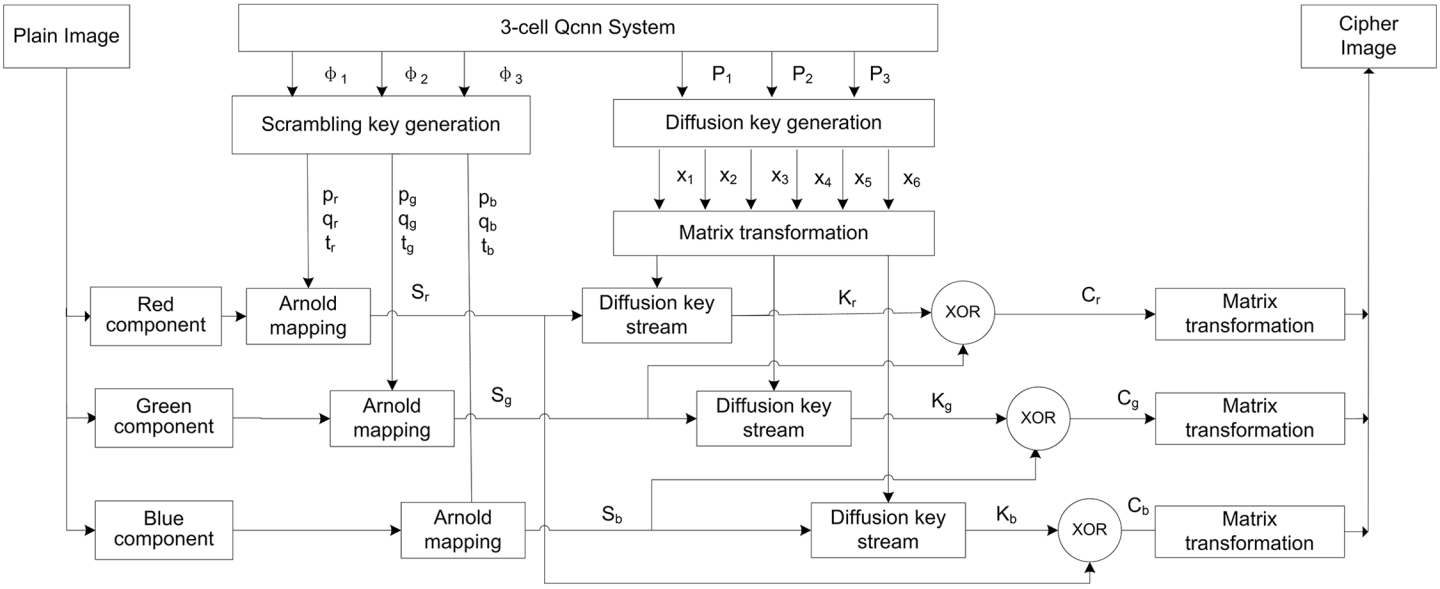

Fig 7. Encryption flowchart.

https://doi.org/10.1371/journal.pone.0184586.g007

three color components of the plain color image.

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = A \begin{pmatrix} x_n \\ y_n \end{pmatrix} \bmod(N) = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} \bmod(N) \tag{10}$$

Since $\det(A) = 1$, the parameters are described as follows:

$$\begin{aligned} p_r &= floor(\bmod(\phi_1 \times 2^{24}), N) \\ q_r &= floor(\bmod(\bmod(\phi_1 \times 2^{48}), 2^{24}), N) \\ p_g &= floor(\bmod(\phi_2 \times 2^{24}), N) \\ q_g &= floor(\bmod(\bmod(\phi_2 \times 2^{48}), 2^{24}), N) \\ p_b &= floor(\bmod(\phi_3 \times 2^{24}), N) \\ q_b &= floor(\bmod(\bmod(\phi_3 \times 2^{48}), 2^{24}), N) \end{aligned}$$

The iterations of Arnold mappings are

$$t_j = floor(((\bmod(\phi_j \times 2^{24}) + \bmod(\phi_j \times 2^{48})), 2^{24}), N) \quad j \in \{r, g, b\} \tag{11}$$

The plain image is scrambled by Eq (10) in order to generate the permutation image. It is transformed into three $1 \times (N \times N)$ streams $S_j = \{S_j(1), S_j(2), \ldots\ldots S_j(N \times N)\}, j \in \{r, g, b\}$ by arranging its pixels from top to bottom and left to right.

In the diffusion stage, 6^{th}-order CNN system(4) is used to diffuse the image, which changes the permutation image pixel's values. The initial conditions are described as follows:

$$x_i(0) = \gamma_i P_j, \qquad (i = 1, 2, 3, 4, 5, 6 \quad j = 1, 2, 3)$$

Of these initial conditions, γ_i is taken as the appropriate integer. P_j is chaotic value, so the initial conditions $x_i(0)$ is also chaotic value. Let the plain image be an N × N image.

The 6^{th}-CNN is iterated $\frac{N\times N}{2}$ times and its result is divided into three matrices:X_r, X_g, and X_b:

$$X_r = \begin{bmatrix} X_1(1) & X_2(1) \\ X_1(2) & X_2(2) \\ \vdots & \vdots \\ X_1\left(\frac{N\times N}{2}\right) & X_2\left(\frac{N\times N}{2}\right) \end{bmatrix}, X_g = \begin{bmatrix} X_3(1) & X_4(1) \\ X_3(2) & X_4(2) \\ \vdots & \vdots \\ X_3\left(\frac{N\times N}{2}\right) & X_4\left(\frac{N\times N}{2}\right) \end{bmatrix},$$

$$X_b = \begin{bmatrix} X_5(1) & X_6(1) \\ X_5(2) & X_6(2) \\ \vdots & \vdots \\ X_5\left(\frac{N\times N}{2}\right) & X_6\left(\frac{N\times N}{2}\right) \end{bmatrix}.$$

Arranging matrix elements from top to bottom and from left to right, X_r, X_g, and X_b are transformed into three $1 \times (N \times N)$ streams:

$$X_j\_Stream(i), \quad (i = 1, 2, ..., N \times N \quad j \in \{r, g, b\})$$

The diffusion key streams, K_j, are generated by using sequences X_j_Stream and S_j as described by Eq (12):

$$\begin{aligned} K_j(i) &= \mathrm{mod}\{round[(abs(X_j\_Stream(i)) - floor(abs(X_j\_Stream(i)))) \times 10^{14} \\ &\quad + S_j(i-1)], N\} \quad i = 1, 2, ..., N \times N, j \in \{r, g, b\} \end{aligned} \tag{12}$$

Let $S_j(0) = 127$. The scramble image is shifted to the cipher image by key streams, K_j.

$$\begin{cases} C_r(i) = bitxor(S_g(i), K_r(i)) \\ C_g(i) = bitxor(S_b(i), K_g(i)) \\ C_b(i) = bitxor(S_r(i), K_b(i)) \end{cases}$$

$i = 1, 2, \ldots\ldots, N \times N$, bitxor(.) function returns the bitwise exclusive OR value of two integers.

These C_r, C_g, and C_b row vectors are transformed into $N \times N$ matrix. Compose the three color components to obtain the encrypted image.

Decryption algorithm

As shown in Fig 8, the decryption is the inverse process of the encryption, except that the decryption key P_{r1}, P_{r2}, P_{r3}, ϕ_{r1}, ϕ_{r2}, and ϕ_{r3} are generated by the synchronized key generation system instead of 3-cell QCNN(2).

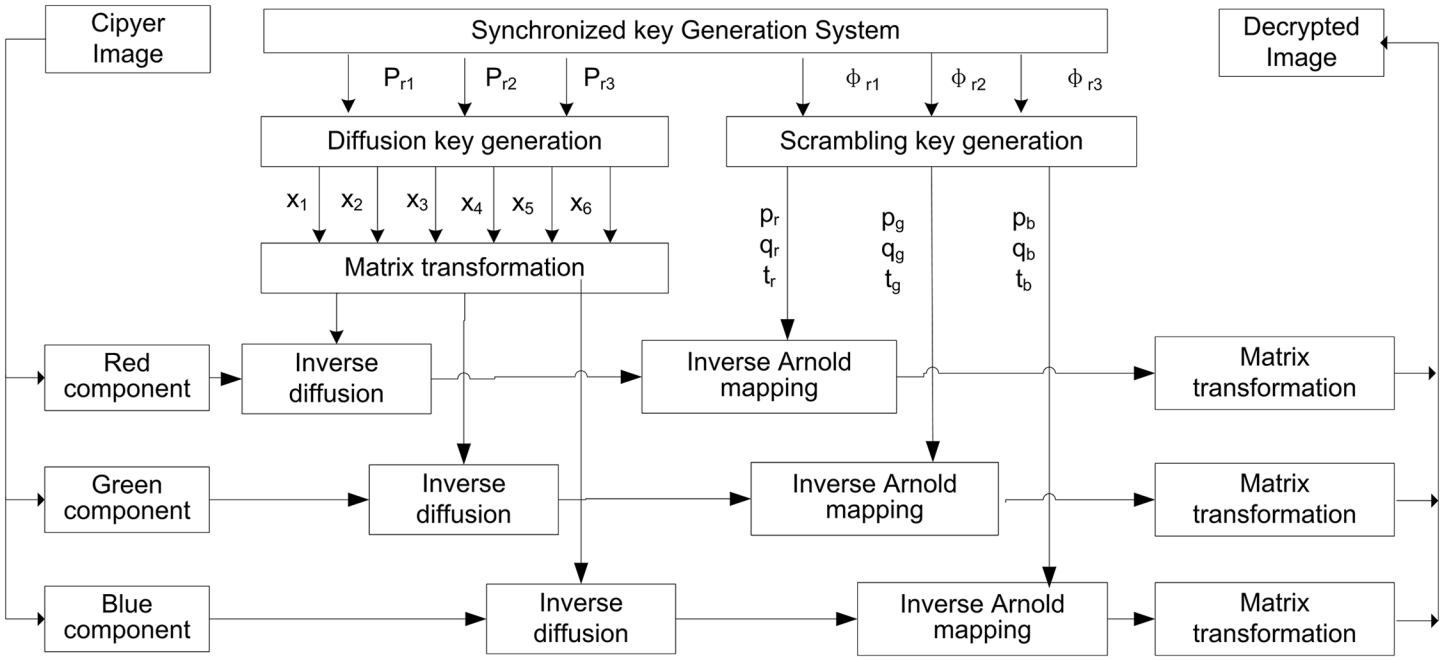

Fig 8. Decryption flowchart.

https://doi.org/10.1371/journal.pone.0184586.g008

Performance analysis

In this section, we perform 11 experiments to validate the proposed scheme. The results show that our scheme has good encryption performance.

Key space analysis

The key space size is the total number of different keys that can be applied in the encryption process. The key space must be large enough to make brute-attacks infeasible. Stinson DR. [38] suggested that the key space should be at least 2^{100} to ensure a high level security. In our algorithm there are twelve parameters for the keys: six initial conditions P_1, Φ_1, P_2, Φ_2, P_3, Φ_3 and six control parameters $b_{01}, b_{02}, b_{03}, \omega_{01}, \omega_{02}, \omega_{03}$. They are all floating point numbers. According to the IEEE floating-point standard [39], the computational precision of the 64-bit double-precision numbers is 2^{-52}. So the key space of the proposed encryption method is $(2^{52})^{12} = 2^{624}$, which is sufficiently large enough to resist all kinds of brute-force attacks.

Key sensitivity analysis

A secure encryption algorithm must be sensitivity to its keys which satisfies the requirement of resisting brute-force attack. Under the same experiment condition as Eq (13). $P_1(0)$, $P_2(0)$, $P_3(0)$, $\phi_1(0)$, $\phi_2(0)$, $\phi_3(0)$ are QCNN system(2) initial conditions, used as user keys in the proposed encryption scheme.

$$Key = \left\{ \begin{array}{l} P_1(0) = -0.131; P_2(0) = -0.135; P_3(0) = -0.123; \\ \phi_1(0) = -184.9; \phi_2(0) = 147.3414; \phi_1(0) = -196.852; \\ b_{01} = b_{01} = b_{01} = 0.28; \\ \omega_{01} = 0.5; \omega_{02} = 0.2; \omega_{03} = 0.3; \end{array} \right\} \quad (13)$$

With a tiny difference in the encryption keys, six groups of test cases are designed, which differ 10^{-13} to every encryption key, respectively.

$$
\begin{aligned}
Key1 &= \left\{ \begin{array}{l} P_1(0) = -0.131 + 10^{-13}; P_2(0) = -0.135; P_3(0) = -0.123; \\ \phi_1(0) = -184.9; \phi_2(0) = 147.3414; \phi_1(0) = -196.852; \\ b_{01} = b_{01} = b_{01} = 0.28; \\ \omega_{01} = 0.5; \omega_{02} = 0.2; \omega_{03} = 0.3; \end{array} \right\} \\
Key2 &= \left\{ \begin{array}{l} P_1(0) = -0.131; P_2(0) = -0.135 + 10^{-13}; P_3(0) = -0.123; \\ \phi_1(0) = -184.9; \phi_2(0) = 147.3414; \phi_1(0) = -196.852; \\ b_{01} = b_{01} = b_{01} = 0.28; \\ \omega_{01} = 0.5; \omega_{02} = 0.2; \omega_{03} = 0.3; \end{array} \right\} \\
Key3 &= \left\{ \begin{array}{l} P_1(0) = -0.131; P_2(0) = -0.135; P_3(0) = -0.123 + 10^{-13}; \\ \phi_1(0) = -184.9; \phi_2(0) = 147.3414; \phi_1(0) = -196.852; \\ b_{01} = b_{01} = b_{01} = 0.28; \\ \omega_{01} = 0.5; \omega_{02} = 0.2; \omega_{03} = 0.3; \end{array} \right\} \\
Key4 &= \left\{ \begin{array}{l} P_1(0) = -0.131; P_2(0) = -0.135; P_3(0) = -0.123; \\ \phi_1(0) = -184.9 + 10^{-13}; \phi_2(0) = 147.3414; \phi_1(0) = -196.852; \\ b_{01} = b_{01} = b_{01} = 0.28; \\ \omega_{01} = 0.5; \omega_{02} = 0.2; \omega_{03} = 0.3; \end{array} \right\} \\
Key5 &= \left\{ \begin{array}{l} P_1(0) = -0.131; P_2(0) = -0.135; P_3(0) = -0.123; \\ \phi_1(0) = -184.9; \phi_2(0) = 147.3414 + 10^{-13}; \phi_1(0) = -196.852; \\ b_{01} = b_{01} = b_{01} = 0.28; \\ \omega_{01} = 0.5; \omega_{02} = 0.2; \omega_{03} = 0.3; \end{array} \right\} \\
Key6 &= \left\{ \begin{array}{l} P_1(0) = -0.131; P_2(0) = -0.135; P_3(0) = -0.123; \\ \phi_1(0) = -184.9; \phi_2(0) = 147.3414; \phi_1(0) = -196.852 + 10^{-13}; \\ b_{01} = b_{01} = b_{01} = 0.28; \\ \omega_{01} = 0.5; \omega_{02} = 0.2; \omega_{03} = 0.3; \end{array} \right\}
\end{aligned}
$$

Table 2 lists the percentage of different pixels in RGB color component using Key or Key1, Key2, . . ., Key6 seven encrypt images, respectively. Therefore, it can be concluded the slightly deviation in the key brings out absolutely different in the corresponding encryption images. Consequently, the proposed scheme has a high key sensitivity and can resist the brute-force attack.

Table 2. Percentage of different pixels in RGB color component using Key or Key1, Key2, . . ., Key6 encrypted images.

Image color component	Key1	Key2	Key3	Key4	Key5	Key6
Airplane Red	99.5956	99.6155	99.5834	99.646	99.5895	99.588
Airplane Green	99.588	99.5911	99.5895	99.5911	99.5911	99.6262
Airplane Blue	99.5743	99.6033	99.6216	99.5941	99.6155	99.5941
Cablecar Red	99.6094	99.6185	99.6231	99.6002	99.6048	99.6353
Cablecar Green	99.5895	99.6017	99.5987	99.5926	99.6170	99.5941
Cablecar Blue	99.5865	99.5972	99.5926	99.6201	99.5880	99.5789
Cornfield Red	99.6567	99.6170	99.6307	99.5972	99.6002	99.5804
Cornfield Green	99.6109	99.6063	99.6109	99.6124	99.5804	99.6323
Cornfield Blue	99.6033	99.5850	99.5895	99.6277	99.5499	99.6063
Peppers Red	99.6124	99.5926	99.6246	99.6078	99.6124	99.6231
Peppers Green	99.6338	99.614	99.588	99.6231	99.5438	99.5804
Peppers Blue	99.6063	99.614	99.5636	99.6033	99.6429	99.5605
Boat Red	99.6017	99.5926	99.5911	99.6048	99.6078	99.588
Boat Green	99.6033	99.6155	99.6399	99.6658	99.5712	99.5834
Boat Blue	99.588	99.5834	99.6246	99.6109	99.588	99.5804
Fruits Red	99.5743	99.5804	99.6384	99.6460	99.6307	99.5865
Fruits Green	99.6201	99.6155	99.5834	99.6506	99.6490	99.5758
Fruits Blue	99.6414	99.6307	99.5712	99.6155	99.6140	99.5987
Yacht Red	99.5880	99.5850	99.6124	99.5911	99.6582	99.6078
Yacht Green	99.5728	99.6521	99.5911	99.6170	99.5758	99.6155
Yacht Blue	99.6521	99.5956	99.6155	99.6429	99.6307	99.5941

https://doi.org/10.1371/journal.pone.0184586.t002

Histogram analysis

A good image encryption approach should always generate the uniform histogram of cipher image for any plain image. The plain images, cipher images, decrypted images, and the histograms of their three-color components are shown in Figs 9–12. As illustrated, the histograms of the encrypted images are fairly uniform and significantly different from the respective histograms of the original images. Hence, our proposed scheme does not provide any clue to statistical attacks.

Correlation coefficient analysis

To test the correlation of pixels (vertical, horizontal, diagonal), we randomly select 4000 adjacent pairs of the plain image and the cipher image, and calculated the correlation coefficients of pixels, according to the following formula:

$$
\begin{aligned}
e(x) &= \frac{1}{N}\sum_{i=1}^{N} x_i \\
d(x) &= \frac{1}{N}\sum_{i=1}^{N} (x_i - e(x))^2 \\
\mathrm{cov}(x,y) &= \frac{1}{N}\sum_{i=1}^{N} (x_i - e(x))(y_i - e(y)) \\
r_{xy} &= \frac{\mathrm{cov}(x,y)}{\sqrt{d(x)}\sqrt{d(y)}}
\end{aligned}
$$

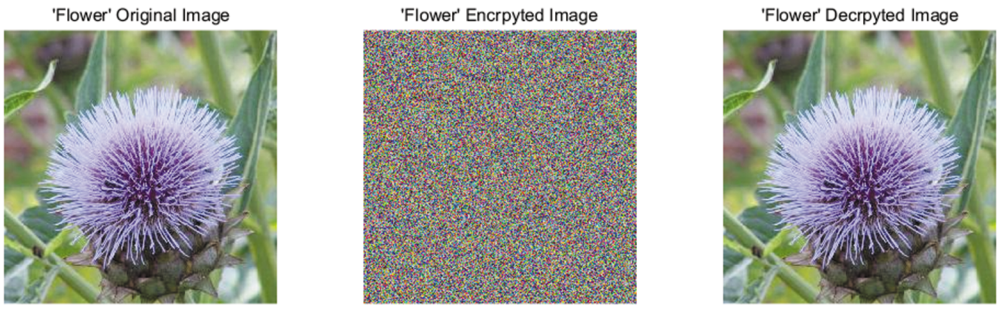

Fig 9. "Flower" original, cipher image and decrypted image.

https://doi.org/10.1371/journal.pone.0184586.g009

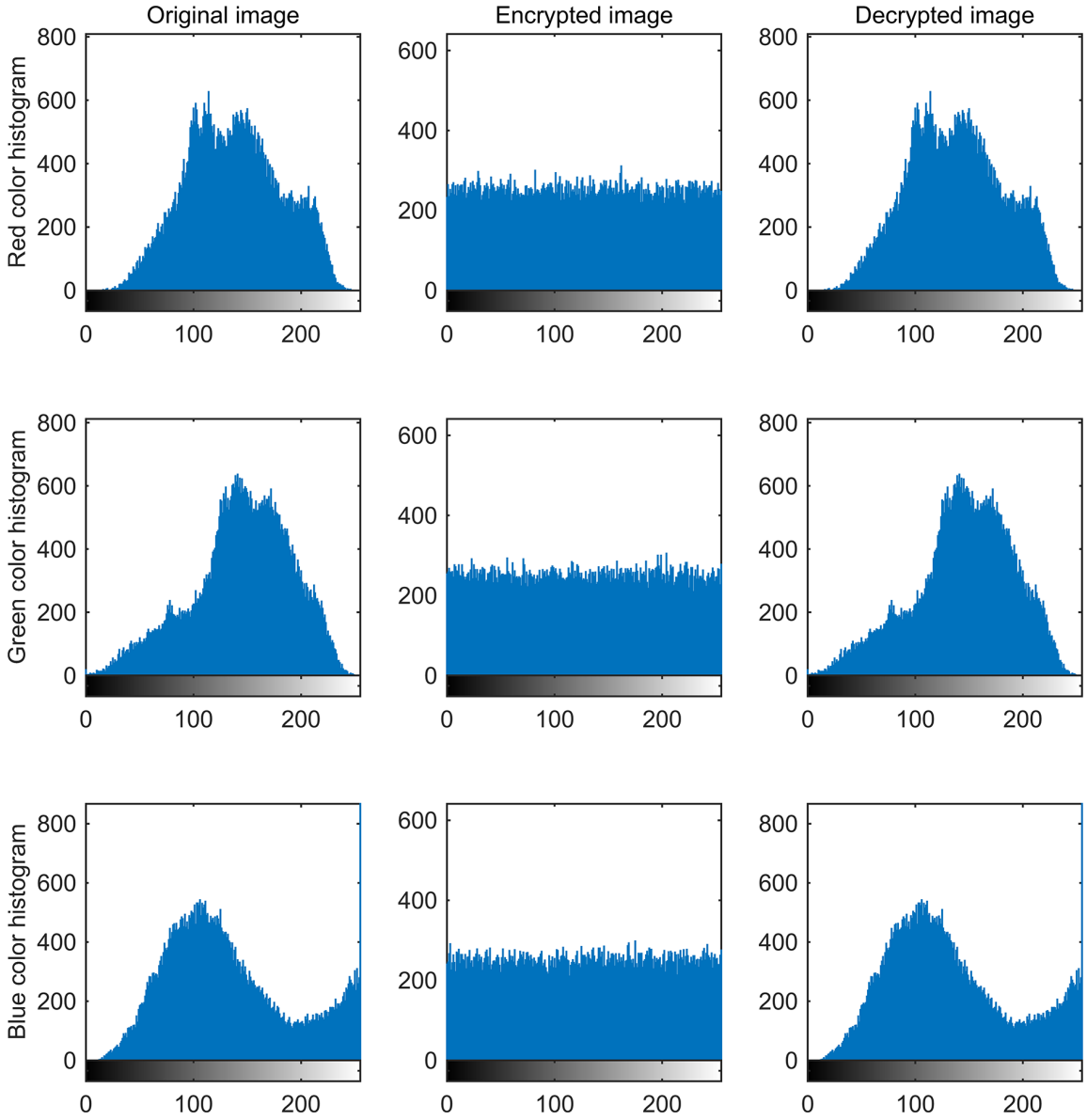

Fig 10. Three color component histograms of "Flower" original, encrypted and decrypted image.

https://doi.org/10.1371/journal.pone.0184586.g010

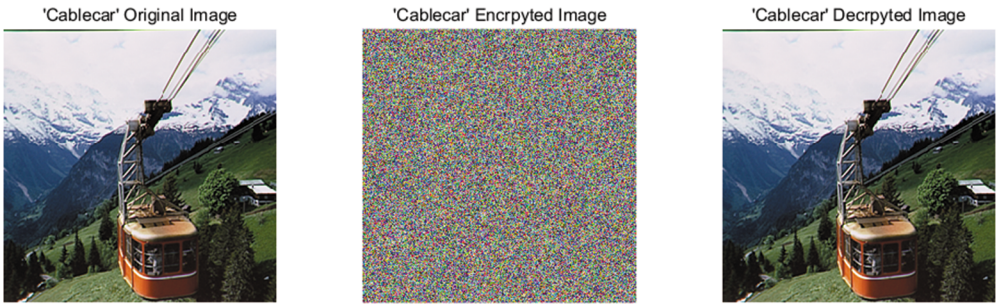

Fig 11. "Cablecar" original, encrypted and decrypted image.

https://doi.org/10.1371/journal.pone.0184586.g011

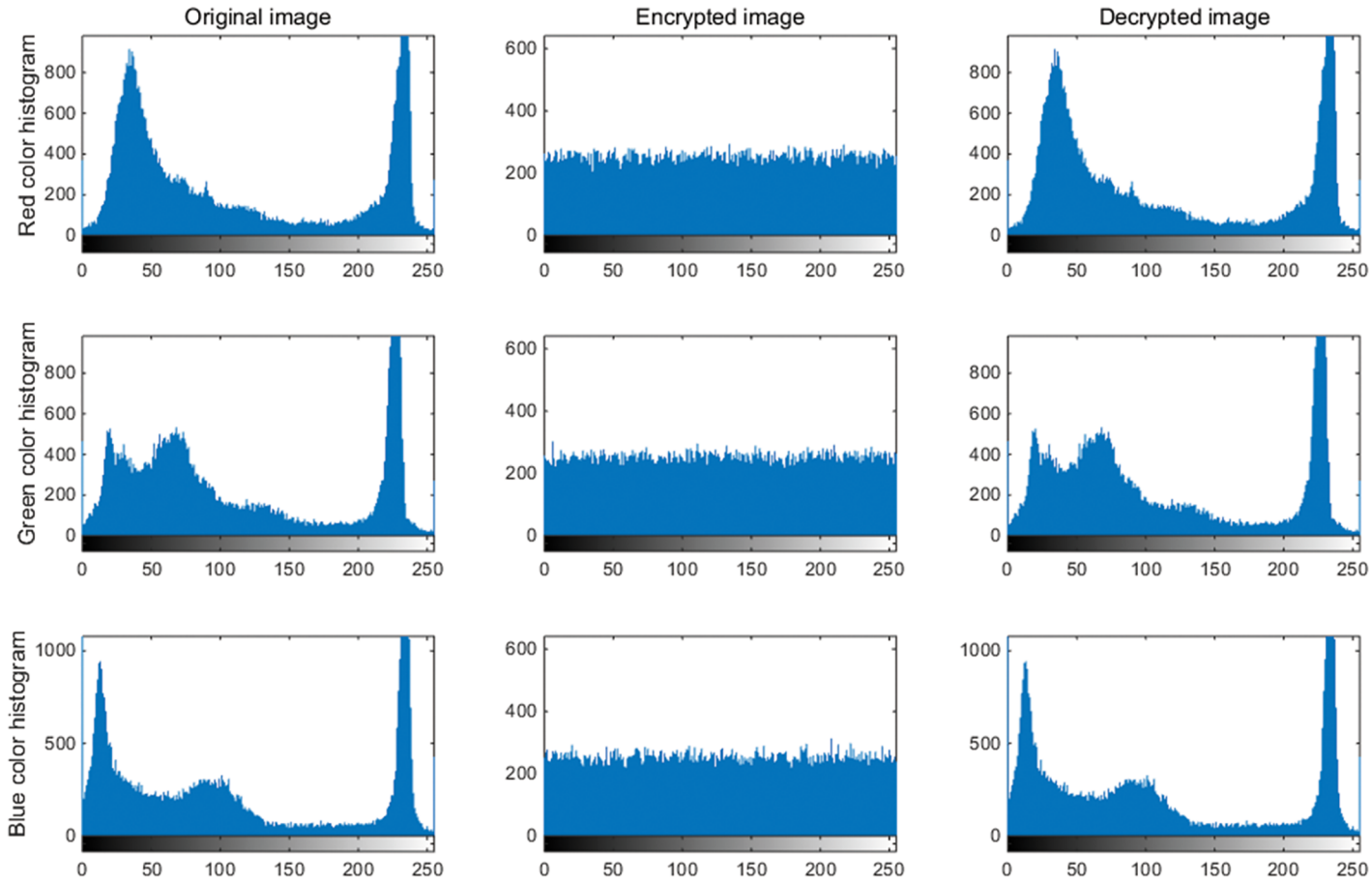

Fig 12. Three color component histograms of "Cablecar" original, encrypted and decrypted image.

https://doi.org/10.1371/journal.pone.0184586.g012

Figs 13 and 14 show image "Flower" and "Cablecar" correlation of two adjacent pixels. Table 3 provides more tests of the correlations, which show that two adjacent pixels in the plain images are highly correlated while the cipher images showed negligible correlations. The result indicates that our proposed encryption model functions properly.

Information entropy analysis. Information entropy is thought to be one of the most important features of randomness. To measure the entropy, $H(m)$, of a source m, the following

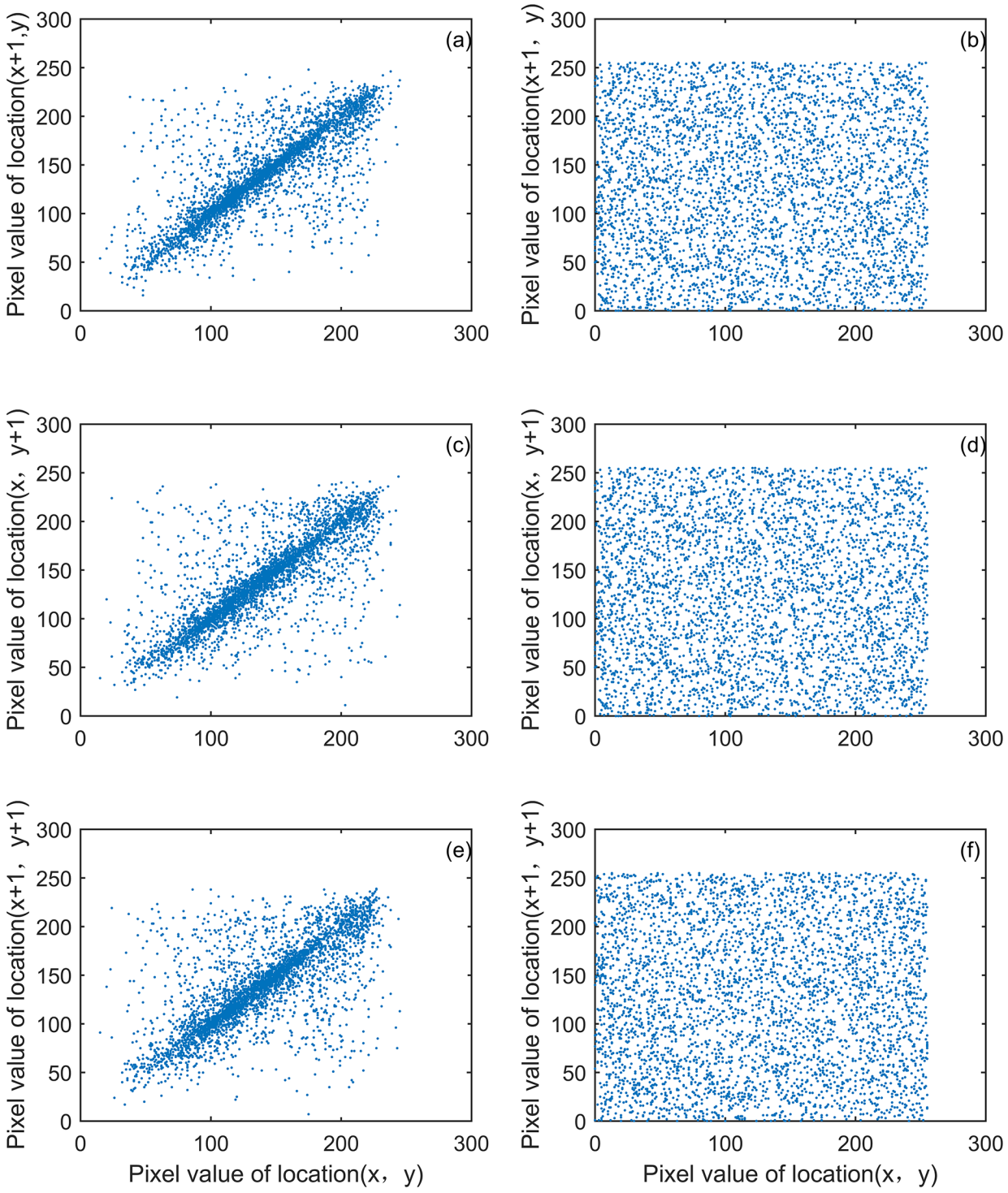

Fig 13. "Flower" image correlation of two adjacent pixels. (a) the distribution of two horizontal adjacent pixels in the original image, (b) the distribution of two horizontal adjacent pixels in the encryption image, (c) the distribution of two vertically adjacent pixels in the original image, (d) the distribution of two vertically adjacent pixels in the encryption image, (e) the distribution of two diagonally adjacent pixels in the original image, and (f) the distribution of two diagonally adjacent pixels in the encryption image.

https://doi.org/10.1371/journal.pone.0184586.g013

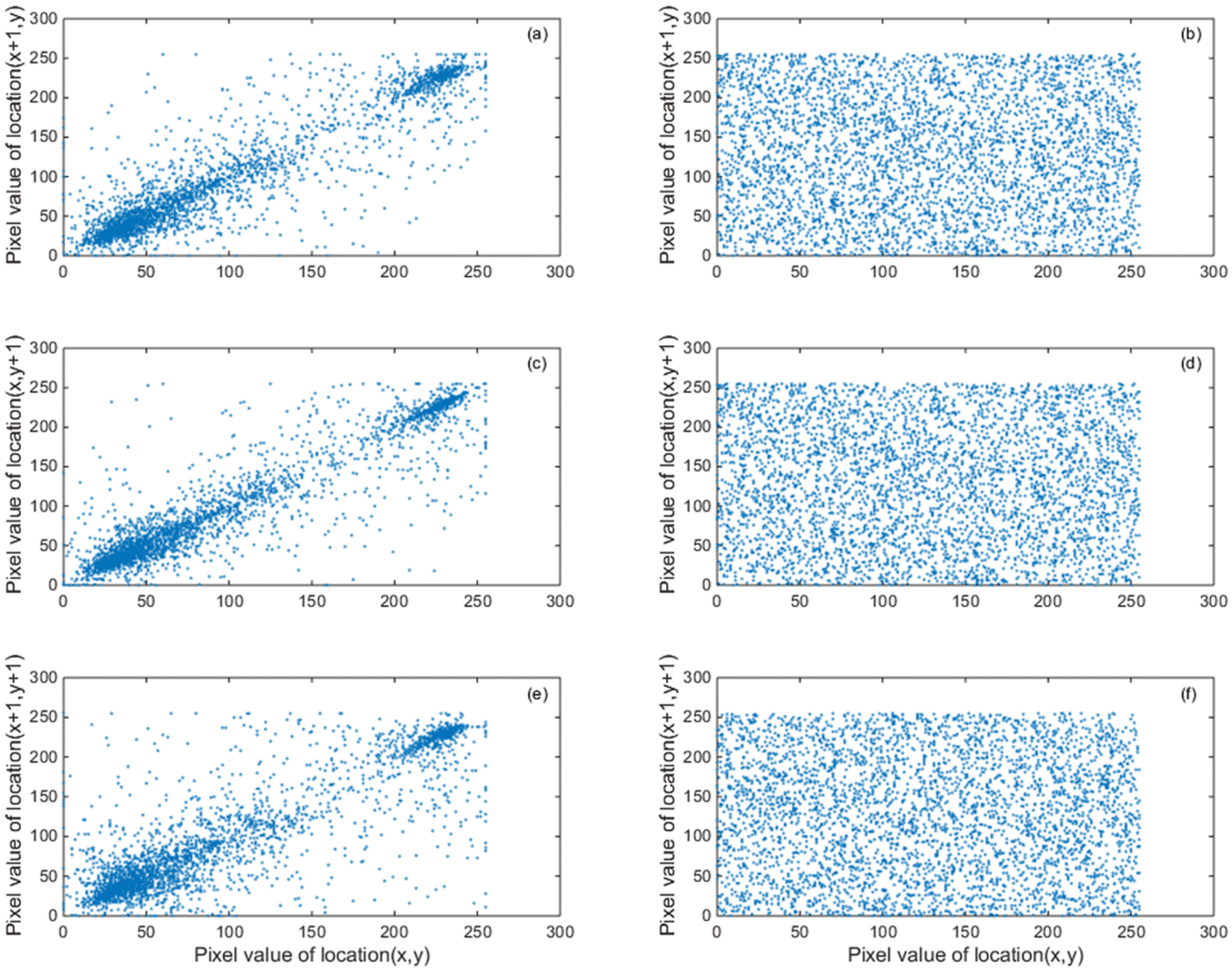

Fig 14. "Cablecar" image correlation of two adjacent pixels. (a) the distribution of two horizontal adjacent pixels in the original image, (b) the distribution of two horizontal adjacent pixels in the encryption image, (c) the distribution of two vertically adjacent pixels in the original image, (d) the distribution of two vertically adjacent pixels in the encryption image, (e) the distribution of two diagonally adjacent pixels in the original image and (f) the distribution of two diagonally adjacent pixels in the encryption image.

https://doi.org/10.1371/journal.pone.0184586.g014

equation can be employed:

$$H(m) = -\sum_{i=0}^{2^n-1} p(m_i) \log_2 p(m_i)$$

where $p(m_i)$ represents the probability of symbol m_i, and the entropy is expressed in bits. For example, when $n = 8$, the image color strength value is $m = \{m_0, \ldots\ldots, m_{255}\}$. For a random process, each symbol has equal probability, $p(m_i) = 1/256$, $H(m) = 8$. In general, the entropy value of the message is smaller than 8 but should to be close to ideal. Table 4 provides a comparison of average entropy values for a considerable number of images for the proposed

Table 3. Correlation coefficients of original images and encryption images.

Encryption algorithm	Horizontal	Vertical	Diagonal
Ref [20] algorithm	0.0681	0.0845	-
Ref [22] algorithm	-0.0318	0.0965	0.0362
Ref [24] algorithm	0.0051	-0.0093	-0.0205
Ref [26] algorithm	0.0086	0.0195	-0.0093
Ref [40] algorithm	-0.00164	0.01304	-0.01911
Ref [41] algorithm	0.0773	0.0770	-0.0.0693
Proposed algorithm "Flower"	-0.0062	0.0052	0.0043
Proposed algorithm "Cablecar"	-0.0061	0.0070	0.0102

https://doi.org/10.1371/journal.pone.0184586.t003

Table 4. Information entropy of ciphered images with three color components.

Encryption algorithm	red	green	blue
Ref [20] algorithm	7.9732	7.9750	7.9715
Ref [22] algorithm	7.9851	7.9852	7.9832
Ref [23] algorithm	7.9991	7.9990	7.9989
Ref [26] algorithm	7.9971	7.9968	7.9974
Ref [41] algorithm	7.9974	7.9966	7.9975
Ref [42] algorithm	7.9808	7.9811	7.9814
Proposed algorithm "Flower"	7.9993	7.9992	7.9987
Proposed algorithm "Cablecar"	7.9972	7.9975	7.9972

https://doi.org/10.1371/journal.pone.0184586.t004

method and some other methods. We noticed that our scheme outperforms other schemes and approaches the ideal value of 8.

Differential attack

Cryptanalysis features an important method called differential attack to crack the encryption algorithm in order to quantitatively measure the influence of a one–pixel change on the cipher image. This influence can be measured via the number of pixel change rate (NPCR) and the unified averaged changing intensity (UACI), which are computed with the following formula:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\%$$

$$UACI = \frac{1}{W \times H}\left(\sum_{i,j}\frac{|C(i,j) - C'(i,j)|}{255}\right) \times 100\%$$

where W and H represent the width and height of the image, respectively. $C(i, j)$ and $C'(i, j)$ are the ciphered images before and after one pixel of the plain image is changed. For position (i, j), if $C(i, j) \neq C'(i, j)$, then $D(i, j) = 1$; else $D(i, j) = 0$. We tested the NPCR and UACI values for images Flower and Cablecar for the proposed scheme. As shown in Tables 5 and 6, the proposed scheme is very sensitive with small changes in the plain image. This result shows that our scheme can resist differential attack well.

Table 5. NPCR of ciphered image.

NPCR	red	green	blue
Ref [23] algorithm	99.6239	99.6216	99.6236
Ref [27] algorithm	99.5445	99.5875	99.5374
Ref [29] algorithm	99.6155	99.6536	99.6475
Ref [41] algorithm	99.6399	99.6002	99.5773
Ref [42] algorithm	99.6170	99.6002	99.5925
Proposed algorithm "Flower"	99.6002	99.6063	99.5834
Proposed algorithm "Cablecar"	99.6140	99.6094	99.5926

https://doi.org/10.1371/journal.pone.0184586.t005

Table 6. UACI of ciphered image.

UACI	red	green	blue
Ref [23]algorithm	33.6623	33.6827	33.6754
Ref [27]algorithm	34.3174	34.1786	33.6467
Ref [29]algorithm	33.6970	34.3251	32.2345
Ref [41]algorithm	33.5916	33.5010	33.4853
Ref [42]algorithm	33.4252	33.5898	33.4466
Proposed algorithm "Flower"	33.3635	33.4891	33.5000
Proposed algorithm "Cablecar"	33.4828	33.2790	33.4992

https://doi.org/10.1371/journal.pone.0184586.t006

Known plaintext attack and chosen plaintext attack

The diffusion key stream K_j, in Eq (12), not only depends on the security key (initial conditions of 3-cell QCNN, $P_1(0)$, $P_2(0)$, $P_3(0)$, $\phi_1(0)$, $\phi_2(0)$, $\phi_3(0)$) but also on the plain image itself. Hence, when the same security key encrypts different images, the diffusion key streams are different. Therefor it is ineffective on input an all "0" or all "1" image into this scheme. Accordingly, our scheme can resist known plaintext attack and chosen plaintext attack.

Encryption quality analysis

In an ideal cryptographic model, encrypted images should have uniform histogram distribution to hide pixels relevant information. It implies the encryption algorithm changes the the cipher pixel value to make the probability of each cipher pixel being totally uniform. Literature [43] gives a method for estimating the encryption quality, deviation from uniform histogram (D_H), which is given by Eq (14).

$$D_H = \frac{\sum_{C_i}^{255} |\mathrm{H}_{C_i} - \mathrm{H}_C|}{M \times N} \tag{14}$$

In Eq (14), $M \times N$ is the image size and C_i is the image pixel gray or color level, $C_i \in [0, 255]$. H_{Ci} is the histogram value at index i, and H_C is the actual histogram of encrypted image. The smaller D_H value indicates the more uniform histogram distribution and the higher encryption quality.

We obtain D_H comparison reports for three images through using our algorithm with other chaotic encryption algorithms in Ref [25]. As can be seen from Table 7, all D_H values are very low. Moreover, our algorithm has more uniform histogram distribution and better encryption quality than Ref [10, 13, 25, 44].

Table 7. Deviation from uniform histogram(D_H).

Image	Proposed algorithm	Ref [10]	Ref [13]	Ref [25]	Ref [44]
Peppers	0.0492	0.0938	0.0979	0.0917	0.0977
Airplane	0.0518	0.0969	0.0995	0.0983	0.0943
Boat	0.0524	0.0902	0.0995	0.0958	0.0985

https://doi.org/10.1371/journal.pone.0184586.t007

Chi-square test

A Chi-squared test [45, 46], also written as X^2 test, is any statistical hypothesis test wherein the sampling distribution of the test statistic is a chi-squared distribution when the null hypothesis is true. Chi-squared test illustrate the possibility of statistical attacks. To evaluate if and what extent distribution of encrypted image histograms approach the features of a uniform distribution, Chi-squared tests are computed for 7 cipher images' histograms, and then are summarized in Table 8. We find the histograms of the encrypted images are fairly uniform, so the proposed scheme can defend statistical attack.

NIST SP800-22 test

NIST SP800-22 test [47] includes 16 test methods, which are used to analyse the randomness of binary sequences generated by cipher systems. We performed all the 16 tests for 65536–8 bits key stream sequence and the results are shown in Table 9. From the Table 9, it shows that our scheme goes through all NIST SP800-22 tests successfully. Therefore, the key stream sequence is absolutely random in our scheme.

Encryption speed and computation complexity

The encryption speed is an important issue for a well applicable encryption system. Nevertheless, it depends on many factors as hardware, software and programming [25]. Ref [24, 48] have performed encryption speed tests for some algorithms in [5, 7, 24, 48–52] at the same enviorment. From Ref [48], we know that the encryption speed of algorithm [5, 7, 48, 49] are >10s, 2.3s, 1.25s, and 2.901s respectively. The execution time of scheme in [24, 50–52] are 155ms, 173ms, 2.089s and 334ms [24]. In our scheme, Arnold mapping iteration times t_j in Eq (11), is randomness for improving security, so it is hard to build a baseline to compare encryption speed with other methods, especially programming skill and code optimization [25]. So we give the encryption speed with different Arnold mapping iteration times in Table 10, and

Table 8. Chi-square test results for encrypted images.

Test Image	X^2 P-value	Decision on H_0
Cablecar	0.710	Accepted
Cornfield	0.969	Accepted
Peppers	0.791	Accepted
Airplane	0.321	Accepted
Fruits	0.580	Accepted
Boat	0.679	Accepted
Yacht	0.684	Accepted

https://doi.org/10.1371/journal.pone.0184586.t008

Table 9. NIST SP800-22 tests results for encrypted key.

Test name		P-value	Result
Frequency		0.9801	Success
Block-frequency		0.2775	Success
Runs		0.3160	Success
Long runs of ones		0.3954	Success
Rank		0.0296	Success
Spectral DFT		0.1550	Success
No overlapping templates		0.9967	Success
Overlapping templates		0.4514	Success
Universal		0.6556	Success
Linear complexity		0.9056	Success
Serial	P-value1	0.9266	Success
Serial	P-value2	0.7865	Success
Approximate entropy		0.6375	Success
Cumulative sums forward		0.5436	Success
Cumulative sums reverse		0.5651	Success
Random excursions	X = -4	0.7220	Success
	X = -3	0.7752	Success
	X = -2	0.2677	Success
	X = -1	0.2656	Success
	X = 1	0.1007	Success
	X = 2	0.3482	Success
	X = 3	0.4977	Success
	X = 4	0.5168	Success
Random excursions variant	X = -9	0.2492	Success
	X = -8	0.1723	Success
	X = -7	0.2026	Success
	X = -6	0.4146	Success
	X = -5	0.4073	Success
	X = -4	0.3178	Success
	X = -3	0.3753	Success
	X = -2	0.6367	Success
	X = -1	0.4315	Success
	X = 1	0.6596	Success
	X = 2	0.7163	Success
	X = 3	0.6525	Success
	X = 4	0.4903	Success
	X = 5	0.3089	Success
	X = 6	0.2110	Success
	X = 7	0.1905	Success
	X = 8	0.1267	Success
	X = 9	0.1269	Success

https://doi.org/10.1371/journal.pone.0184586.t009

Table 10. The speed range for the proposed algorithm.

Image	Iteration 1 time Speed(ms)	Iteration 10 times Speed(ms)	Iteration 20 times Speed(ms)	Iteration 30 times Speed(ms)	Iteration 40 times Speed(ms)	Iteration 50 times Speed(ms)
Peppers	103	368	666	960	1261	1554
Cablecar	102	359	644	931	1215	1501
Airplane	104	372	666	959	1256	1550
Cornfield	101	358	646	934	1214	1512
Boat	101	371	668	960	1256	1553
Fruits	102	363	655	931	1216	1528
Yacht	101	358	644	932	1215	1504

https://doi.org/10.1371/journal.pone.0184586.t010

the environment is Microsoft Windows 7, Matlab8.4, a laptop with an Intel Xeon CPU E3-1220 v3 3.10GHz, 8.00GB RAM. As can be seen from the Table 10, our scheme has an acceptable speed.

Additionally, the computation complexity relies on the number of operations and steps to fulfill the encryption. Our scheme needs $\mathcal{O}(n)$ to complete the entire encryption process, where n is the pixel number of images. Thus, the efficiency of the proposed algorithm is competent in the application level encryption requirements.

Conclusion

In this paper, a semi-symmetric image encryption scheme based on function projective synchronization between two hyperchaotic systems is proposed, and it has several advantages such as great speed, relatively low complexity compared respectively to symmetric and asymmetric algorithms. Especially, the key is generated simultaneously in encryption side and decryption side independently, which effectively avoids the key transmission and threats of key exposure. The presented scheme is a hybrid chaotic encryption algorithm and it consists of a scrambling stage and a diffusion stage. Moreover, the 6^{th}-order CNN is not only regarded as the drive system for the key synchronization, but also is used for diffusing key generation to enhance the security and sensitivity of the scheme. The simulation experiments and security performance analyses show that our scheme has a satisfactory security performance.

Supporting information

S1 Fig. "Flower" original image.
(TIF)

S2 Fig. "Cablecar" original image.
(TIF)

S3 Fig. "Airplane" original image.
(TIF)

S4 Fig. "Boat" original image.
(TIF)

S5 Fig. "Cornfield" original image.
(TIF)

S6 Fig. "Fruits" original image.
(TIF)

S7 Fig. "Peppers" original image.
(TIF)

S8 Fig. "Yacht" original image.
(TIF)

Acknowledgments

This research is partially supported by Industrial Innovation Project of Jilin Province (2016C087, http://jldrc.gov.cn) and Science and Technology Project of Jilin Province (20150312030ZX, http://www.jlkjt.gov.cn/bsfw/kjjhxmsb/).

Author Contributions

Conceptualization: Jinqing Li.

Data curation: Jinqing Li.

Formal analysis: Jinqing Li.

Funding acquisition: Xiaoqiang Di.

Investigation: Jinqing Li.

Methodology: Jinqing Li.

Project administration: Xiaoqiang Di, Huamin Yang.

Resources: Hui Qi, Ligang Cong.

Software: Hui Qi, Ligang Cong.

Supervision: Xiaoqiang Di, Huamin Yang.

Validation: Hui Qi, Ligang Cong.

Visualization: Hui Qi, Ligang Cong.

Writing – original draft: Jinqing Li.

Writing – review & editing: Xiaoqiang Di.

References

1. Matthews R. On the derivation of a "Chaotic" encryption algorithm. Cryptologia. 1989; 8(8):29–41. https://doi.org/10.1080/0161-118991863745
2. Zhang YQ, Wang XY. A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice. Information Sciences. 2014; 273(8):329–51. https://doi.org/10.1016/j.ins.2014.02.156
3. Zhang Q, Guo L, Wei X. Image encryption using DNA addition combining with chaotic maps. Mathematical & Computer Modelling. 2010; 52(11):2028–35. https://doi.org/10.1016/j.mcm.2010.06.005
4. Ji WY, Kim H. An image encryption scheme with a pseudorandom permutation based on chaotic maps. Communications in Nonlinear Science & Numerical Simulation. 2010; 15(12):3998–4006. https://doi.org/10.1016/j.cnsns.2010.01.041
5. Ye R. A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism. Optics Communications. 2011; 284(22):5290–8. https://doi.org/10.1016/j.optcom.2011.07.070
6. Ye G, Wong KW. Wong K-W: An efficient chaotic image encryption algorithm based on a generalized Arnold map. Nonlinear Dyn. 69, 2079–2087. Nonlinear Dynamics. 2012; 69(4):2079–87. https://doi.org/10.1007/s11071-012-0409-z
7. Ye G. Image scrambling encryption algorithm of pixel bit based on chaos map. Pattern Recognition Letters. 2010; 31(5):347–54. https://doi.org/10.1016/j.patrec.2009.11.008

8. Wang Y, Wong KW, Liao X, Chen G. A new chaos-based fast image encryption algorithm. Applied Soft Computing. 2011; 11(1):514–22. https://doi.org/10.1016/j.asoc.2009.12.011
9. Wang X, Zhao J, Liu H. A new image encryption algorithm based on chaos. Optics Communications. 2012; 285(5):562–6. https://doi.org/10.1016/j.optcom.2011.10.098
10. Wang X, Teng L, Qin X. A novel colour image encryption algorithm based on chaos. Signal Processing. 2012; 92(4):1101–8. https://doi.org/10.1016/j.sigpro.2011.10.023
11. Wang X. An image blocks encryption algorithm based on spatiotemporal chaos. Nonlinear Dynamics. 2012; 67(1):365–71. https://doi.org/10.1007/s11071-011-9984-7
12. Wang X, Luan D. A novel image encryption algorithm using chaos and reversible cellular automata. Communications in Nonlinear Science & Numerical Simulation. 2013; 18(11):3075–85. https://doi.org/10.1016/j.cnsns.2013.04.008
13. Wang X, Liu L, Zhang Y. A novel chaotic block image encryption algorithm based on dynamic random growth technique. Optics & Lasers in Engineering. 2015; 66(66):10–8. https://doi.org/10.1016/j.optlaseng.2014.08.005
14. Wang Xing-yuan, Chen Feng, Wang Tian. A new compound mode of confusion and diffusion for block encryption of image based on chaos. Communications in Nonlinear Science & Numerical Simulation. 2010; 15(9):2479–85. https://doi.org/10.1016/j.cnsns.2009.10.001
15. Volos CK, Kyprianidis IM, Stouboulos IN. Image encryption process based on chaotic synchronization phenomena. Signal Processing. 2013; 93(5):1328–40. https://doi.org/10.1016/j.sigpro.2012.11.008
16. Seyedzadeh SM, Mirzakuchaki S. A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. Signal Processing. 2012; 92(5):1202–15. https://doi.org/10.1016/j.sigpro.2011.11.004
17. Patidar V, Pareek NK, Purohit G, Sud KK. A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption. Optics Communications. 2011; 284(19):4331–9. https://doi.org/10.1016/j.optcom.2011.05.028
18. Mirzaei O, Yaghoobi M, Irani H. A new image encryption method: parallel sub-image encryption with hyper chaos. Nonlinear Dynamics. 2011; 67(1):557–66. https://doi.org/10.1007/s11071-011-0006-6
19. Liu L, Zhang Q, Wei X. A RGB image encryption algorithm based on DNA encoding and chaos map: Pergamon Press, Inc. 2012; 38(5): 1240–8.
20. Rhouma R, Meherzi S, Belghith S. OCML-based colour image encryption. Chaos Solitons & Fractals. 2009; 40(1):309–18. https://doi.org/10.1016/j.chaos.2007.07.083
21. Fu C, Lin BB, Miao YS, Liu X, Chen JJ. A novel chaos-based bit-level permutation scheme for digital image encryption. Optics Communications. 2011; 284(23):5415–23. https://doi.org/10.1016/j.optcom.2011.08.013
22. Liu H, Wang X. Color Image Encryption based on One-Time Keys and Robust Chaotic Maps. Computers & Mathematics with Applications. 2010; 59(10):3320–7. https://doi.org/10.1016/j.camwa.2010.03.017
23. Belazi A, Khan M, El-Latif AAA, Belghith S. Efficient cryptosystem approaches: S-boxes and permutation–substitution-based encryption. Nonlinear Dynamics. 2016:1–25.
24. Belazi A, El-Latif AAA, Diaconu AV, Rhouma R, Belghith S. Chaos-based partial image encryption scheme based on linear fractional and lifting wavelet transforms. Optics & Lasers in Engineering. 2017; 88:37–50. https://doi.org/10.1016/j.optlaseng.2016.07.010
25. Belazi A, El-Latif AAA, Belghith S. A novel image encryption scheme based on substitution-permutation network and chaos. Signal Processing. 2016; 128:155–70. https://doi.org/10.1016/j.sigpro.2016.03.021
26. Li J.Q BFM, Di XQ. New color image encryption algorithm based on compound chaos mapping and hyperchaotic cellular neural network. Journal of Electronic Imaging. 2013; 22(1):3036.
27. Tong XJ. Design of an image encryption scheme based on a multiple chaotic map. Communications in Nonlinear Science & Numerical Simulation. 2013; 18(7):1725–33. https://doi.org/10.1016/j.cnsns.2012.11.002
28. Zhang Y, Xiao D. An image encryption scheme based on rotation matrix bit-level permutation and block diffusion. Communications in Nonlinear Science & Numerical Simulation. 2014; 19(1):74–82. https://doi.org/10.1016/j.cnsns.2013.06.031
29. El-Latif AAA, Li L, Wang N, Han Q, Niu X. A new approach to chaotic image encryption based on quantum chaotic system, exploiting color spaces. Signal Processing. 2013; 93(11):2986–3000. https://doi.org/10.1016/j.sigpro.2013.03.031
30. Cheng CJ, Cheng CB. An asymmetric image cryptosystem based on the adaptive synchronization of an uncertain unified chaotic system and a cellular neural network. Communications in Nonlinear Science & Numerical Simulation. 2013; 18(10):2825–37. https://doi.org/10.1016/j.cnsns.2013.02.011

31. Stallings W. Cryptography and Network Security: Principles and Practice. International Journal of Engineering & Computer Science. 1999; 01(01):121–36.
32. Carroll TL, Pecora LM, Heagy JF. Synchronization of nonautonomous chaotic systems. Patent Application Department of the Navy, Washington, DC.; 1995.
33. Toth G, Lent CS, Tougaw PD, Brazhnik Y, Weng W, Porod W, et al. Quantum cellular neural networks. Superlattices & Microstructures. 2000; 20(4):473–8. https://doi.org/10.1006/spmi.1996.0104
34. Wang X, Bing X, Zhang H. A multi-ary number communication system based on hyperchaotic system of 6th-order cellular neural network. Communications in Nonlinear Science & Numerical Simulation. 2010; 15(1):124–33. https://doi.org/10.1016/j.cnsns.2009.03.035
35. Sudheer KS, Sabir M. Adaptive function projective synchronization of two-cell Quantum-CNN chaotic oscillators with uncertain parameters. Physics Letters A. 2009; 373(21):1847–51. https://doi.org/10.1016/j.physleta.2009.03.052
36. Fortuna L. QUANTUM-CNN TO GENERATE NANOSCALE CHAOTIC OSCILLATORS. International Journal of Bifurcation & Chaos. 2004; 14(03):1085–9. https://doi.org/10.1142/S0218127404009624
37. Bernhard MA. Introduction to Chaotic Dynamical Systems. Introduction to Chaotic Dynamical Systems. 1992.
38. Stinson DR. Cryptography: Theory and practice. 3rd ed: Unesco; 2006. 65 p.
39. IEEE, editor IEEE Standard for Floating-Point Arithmetic. IEEE Std; 2008.
40. Al-Mashhadi HM, Abduljaleel IQ, editors. Color image encryption using chaotic maps, triangular scrambling, with DNA sequences. 2017 International Conference on Current Research in Computer Science and Information Technology (ICCIT); 2017.
41. Guesmi R, Farah MAB, Kachouri A, Samet M. Hash key-based image encryption using crossover operator and chaos. Multimedia Tools & Applications. 2016; 75(8):4753–69. https://doi.org/10.1007/s11042-015-2501-0
42. Liu H, Kadir A, Gong P. A fast color image encryption scheme using one-time S-Boxes based on complex chaotic system and random noise. Optics Communications. 2015; 338:340–7. https://doi.org/10.1016/j.optcom.2014.10.021
43. Elashry I, Faragallah O, Abbas A, El-Rabaie S, El-Samie FA. A New Method for Encrypting Images with Few Details Using Rijndael and RC6 Block Ciphers in the Electronic Code Book Mode. Information Security Journal A Global Perspective. 2012; 21(4):193–205. https://doi.org/10.1080/19393555.2011.654319
44. Hua Z, Zhou Y, Pun CM, Chen CLP. 2D Sine Logistic modulation map for image encryption. Information Sciences. 2015; 297:80–94. https://doi.org/10.1016/j.ins.2014.11.018
45. Gagunashvili ND. CHICOM: A code of tests for comparing unweighted and weighted histograms and two weighted histograms. Computer Physics Communications. 2012; 183(1):193–6. https://doi.org/10.1016/j.cpc.2011.08.014
46. Gagunashvili ND. CHIWEI: A code of goodness of fit tests for weighted and unweighted histograms. Computer Physics Communications. 2012; 183(2):418–21. https://doi.org/10.1016/j.cpc.2011.10.009
47. Pareschi F, Rovatti R, Setti G. On Statistical Tests for Randomness Included in the NIST SP800-22 Test Suite and Based on the Binomial Distribution. IEEE Transactions on Information Forensics & Security. 2012; 7(2):491–505. https://doi.org/10.1109/TIFS.2012.2185227
48. Wu X, Li Y, Kurths J. A New Color Image Encryption Scheme Using CML and a Fractional-Order Chaotic System. Plos One. 2014; 10(3):e0119660. https://doi.org/10.1371/journal.pone.0119660
49. Abdullah AH, Enayatifar R, Lee M. A hybrid genetic algorithm and chaotic function model for image encryption. AEUE—International Journal of Electronics and Communications. 2012; 66(10):806–16. https://doi.org/10.1016/j.aeue.2012.01.015
50. ur Rehman A, Liao X, Kulsoom A, Abbas SA. Selective encryption for gray images based on chaos and DNA complementary rules. Multimedia Tools and Applications. 2015; 74(13):4655–77. https://doi.org/10.1007/s11042-013-1828-7
51. Zhou Y, Cao W, Philip Chen CL. Image encryption using binary bitplane. Signal Processing. 2014; 100:197–207. https://doi.org/10.1016/j.sigpro.2014.01.020
52. Hsiao H-I, Lee J. Fingerprint image cryptography based on multiple chaotic systems. Signal Process. 2015; 113(C):169–81. https://doi.org/10.1016/j.sigpro.2015.01.024