Nonlinear Dyn (2015) 80:1271–1290
DOI 10.1007/s11071-015-1943-2

ORIGINAL PAPER

# A cycling chaos-based cryptic-free algorithm for image steganography

**Mahdi Aziz · Mohammad H. Tayarani-N · Mehdi Afsar**

Received: 20 July 2014 / Accepted: 28 January 2015 / Published online: 11 February 2015


**Abstract** In recent years, chaotic systems have surfaced to become an important field in steganographic matters. In this paper, we present a simple cryptic-free least significant bits spatial- domain-based steganographic technique that embeds information (a color or a grayscale image) into a color image. The proposed algorithm, called cycling chaos-based steganographic algorithm, comprises two main parts: A cycling chaos function that is used for generating the seeds for pseudorandom number generator (PRNG) and PRNG that is utilized for determining the channel and the pixel positions of the host image in which the sensitive data are stored. The proposed algorithm is compared with two powerful steganographic color image methods in terms of peak signal-to-noise ratio and quality index. The comparisons indicate that the proposed algorithm shows good hiding capacity and fulfills stego-image quality. We also compare our algorithm against some existing steganographic attacks including RS attack, Chi-square test, byte attack and visual attack. The results demonstrate that the proposed algorithm can successfully withstand against these attacks.

**Keywords** Chaotic methods · Cycling chaos · Steganography · Image hiding

M. Aziz (✉)
AmirKabir University of Technology, Vali Asr St., Tehran, Iran
e-mail: mahdi_aziz@aut.ac.ir

M. H. Tayarani-N
Department of Computer Science, University of Birmingham, Birmingham, UK

M. Afsar
Sama Technical and Vocational Training College, Islamic Azad University, Mashhad Branch, Mashhad, Iran
e-mail:m.afsar@qiau.ac.ir

## 1 Introduction

As sensitive information transferred among large computer networks increases, information security is becoming one of the integral parts of network infrastructure in recent years. Cryptography as an important tool to encrypt sensitive information in the networks has been the mainstay of information security for a long time. However, the problem of using cryptographic techniques is that they attract malicious attacker's interests. That is when a cryptographic technique is used, as the encrypted secret data are meaningless, attackers or intruders know that there is sensitive data in the carrier media [1–5]. Consequently, by using powerful computer systems or grid computers, they can decrypt the data. To cope with this, information hiding techniques are used. Information hiding or "steganography" refers to imperceptibly concealing information in a dummy container where no one except the sender and the receiver of the transmitted information knows that the data are embedded. The innocent-looking media used to cover the sensitive information is called the host image, and when the data are hidden in the media, it is called the stego-image.

Taking advantage of less sensitive human visual system, image hiding can be considered as a good choice for camouflaging important information. There are a variety of image hiding techniques. The substitution technique is one of them that has gained widespread attentions because of its simplicity and ease of use [2,4,6,7]. This technique is to manipulate the bit plane of the carrier image by directly replacing some bits of the image with sensitive information. This is a very useful technique as there are some bits of the host image pixels that are redundant and can be replaced by the secret bit data [3]. If these bits are set in the LSBs of host image pixels, the approach is called LSBs substitution technique, and if they are placed in the most significant bits of host image pixels, the technique is called MSBs substitution [3].

Depending on how the secret data are embedded in the host image, steganographic methods can be categorized into two main groups: spatial-domain- and transform-based algorithms [3]. Spatial- domain-based algorithms usually conceal the secret data in the LSB plane of the host image while the transform-based algorithms hide the secret message in the significant areas of the host image.

Chaos systems are usually known to be sensitive to their control parameters and initial conditions. They are also recognized as unpredictable, pseudorandom signals that have ergodicity and mixing properties [8,9]. Numerous chaos-based steganographic algorithms have been developed in recent years. For example, [3] proposed a new LSB spatial-domain steganographic algorithm which is based on a chaotic map. The paper utilizes the two-dimensional Arnold cat chaos map for determining the pixel position of the host image in which the secret data are embedded. In another work, [10] uses a simple logistic map for determining the block of the host image in which the sensitive data are sequentially hidden.

A respectable steganographic algorithm should meet some requirements like having high capacity, having high quality, being resistant against steganographic attacks and being cryptic-free. Fulfilling all these requirements, the proposed algorithm is a new LSBs spatial-domain steganographic algorithm which is based on a combination of cycling chaos system and PRNG. Both cycling chaos system and PRNG are employed as the building blocks in designing the proposed algorithm. In the proposed algorithm, the cycling chaos signal is used for producing the seeds for PRNG. Because the seeds need to be deterministic and pseudorandom, the cycling chaos system can be a very good choice. A general version of the cycling chaos signal is a symmetric combination of three interconnected signal components that cover all the domain of the chaotic function, where one signal is active, but the other two complementary signal components are quiescent. The cycling chaos system has eight parameters that can be utilized as the key in the transmission. Since the key space of the cycling chaos system is very large, it is useful in creating the seeds of the PRNG. The PRNG, on the other hand, is very powerful in generating the pseudorandom sequences of pixel positions [11]. In the proposed algorithm, the secret data are embedded in the LSB plane of all color pixels of the host image in a psedorandom-like scheme. To show the significance of the proposed algorithm, we compare it with Yu et al. method [5] and Lin et al.s method [12] on both grayscale and color images. The algorithm is also exposed to RS attack, byte attack, Chi-square test and visual attack. The results demonstrate that the proposed algorithm is a better steganographic algorithm in terms of quality of the stego-image, the resistance against steganoanalysis attacks and hiding capacity.

The rest of this paper is organized as follows. In Sect. 2, the background for the proposed algorithm is presented. In Sect. 3, the model for embedding and extracting secret messages is suggested. In Sect. 4, the proposed algorithm is presented and discussed. In Sect. 5, the results of comparison between the proposed algorithm and some existing methods are reported. And finally, Sect. 6 concludes the paper.

## 2 Background

In this section, we give a short review on color quantization and cycling chaos systems that are widely used in data hiding.

### 2.1 Color quantization

Quantization refers to a lossy compression technique that is achieved by reducing a range of values to a single quantum value. Color images usually consist of 16 million colors embedded into 24-bits image pixels. In some applications like image processing and steganog-

raphy [5], it is preferred to lower the number of image colors in order to make the image easier to store, transmit or display. There are various ways for color quantization that can be classified into two classes: preclustering (hierarchical clustering) methods and postclustering (partitional clustering) methods [13]. Recursively finding nested clusters in a bottom-up or top-down manners, the preclustering methods are based on the statistical analysis of the color distribution of images. Examples of this class are binary splitting [14], median-cut [15], variance-based methods [16] and the recent methods proposed in [17–19].

The postclustering methods, on the other hand, do not find clusters in a hierarchical scheme; instead, they find all the clusters at the same time. Compared to preclustering methods, these methods usually offer higher quality results, but compromise the computational time factor [19]. Examples of this class are found in [20,21].

In this paper, we use the minimum variance-based preclustering color quantization technique in which the color image cube is divided into a large number of smaller boxes, and then, each color located in each box is mapped onto the color value of the center of the box. For more information, see [22].

### 2.2 Pseudorandom number generator

In order to generate pseudorandom numbers, PRNG is used. PRNG uses the seeds provided by external sources (i.e., dynamic functions) to generate pseudorandom numbers. The seed value of PRNG should be unpredictable and random. Therefore, it is often obtained from a random number generator. The outputs of the PRNG are called pseudorandom numbers because they are deterministic values, and each number of pseudorandom sequence can be generated from the seed values.

### 2.3 Cycling chaos systems

Cycling chaos systems [8] are chaotic systems that are proposed to enhance the security of chaotic-based cryptographic systems [9]. Like other chaotic systems, the cycling chaos system has all the required chaotic properties including sensitivity to its initial conditions and control parameters, ergodicity and mixing properties. It is called cycling because the chaotic behavior is circularly shifted among its signal components. This means that when one component of the cycling chaos signal is active, the other complementary components are inactive. A $N$ k-dimensional chaotic system is described as

$$x_{i\,n+1} = f(x_{i\,n}, \lambda_i)\text{n},$$

where $x_i = (x_{i\,1}, x_{i\,2}, ..., x_{i\,k}) \subset R^k$ denotes the cycling chaos signal.

In general, the cycling chaos system is defined as a three-dimensional signal.

$$\begin{aligned} x_{n+1} &= \lambda_1 x_n - x_n^3 - \gamma |y_n|^m x_n, \\ y_{n+1} &= \lambda_2 y_n - y_n^3 - \gamma |z_n|^m y_n, \\ z_{n+1} &= \lambda_3 z_n - z_n^3 - \gamma |x_n|^m z_n, \end{aligned}$$

where $\lambda_1$, $\lambda_2$ and $\lambda_3$ are the internal conditions and $\gamma$ and $m$ are the control parameters.

Figure 1 exhibits the time series of the cycling chaos signals where $x_0 = -0.01, y_0 = 0.03, z_0 = 0.02, \lambda_1 = 3.0, \lambda_2 = 2.98, \lambda_3 = 2.87, \gamma = 3.05$ and $m = 0.25$.

As shown in Fig. 1, each component of cycling chaos forms a different part of the cycling chaos signal, covering all the domain of the signal. In Fig. 1d, the aggregation of the cycling chaos signal components is shown. As shown in Fig. 1d, the cycling chaos signal is unpredictable and irregular.

## 3 Proposed model

In this paper, we use the secret key stego-system with embedding/extracting model. This model is utilized by many existing steganographic software and is developed by several research papers [23–25]. Figure 2 shows the schematic of the proposed steganographic system, which is based on the information-theoretic model [23,24].

In this system, there are three involving entities: sender that sends the stego-image, receiver that receives the transmitted stego-image and intruder that illegally attempts to hear and understand the data. The intruder is able to read all the data that is transmitted through the public channel. The channel of transmission is asynchronous, and the data therein can be transmitted sequentially (one bit at a time). The order of data is unchanged during the transmission.

In this model, there are two different types of transmission media: private and public channels. Each of

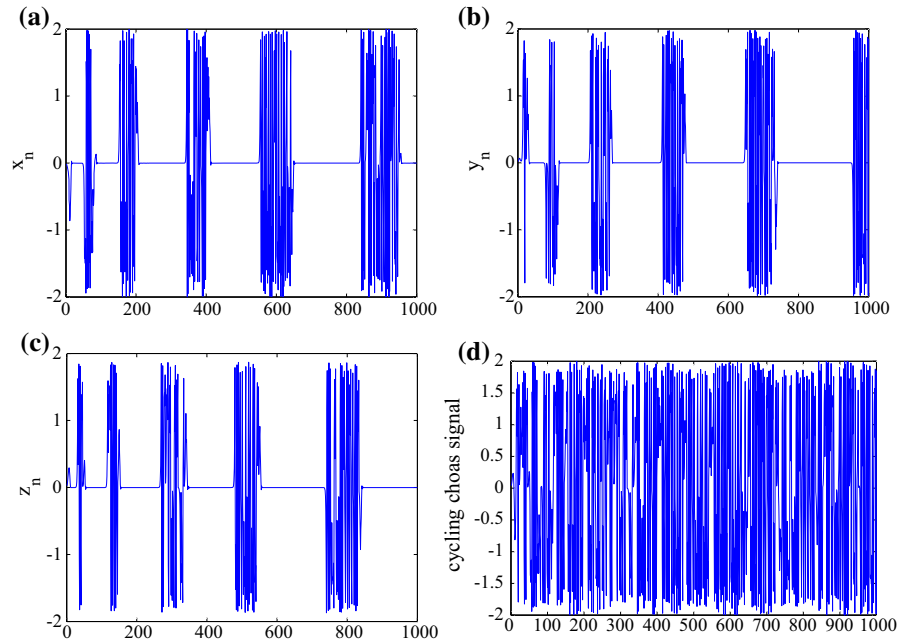


**Fig. 1** The time series of cycling chaos signal where $x_0 = -0.01$, $y_0 = 0.03$, $z_0 = 0.02$, $\lambda_1 = 3.0$, $\lambda_2 = 2.98$, $\lambda_3 = 2.87$, $\gamma = 3.05$ and $m = 0.25$. **a** The time series of $x_n$, **b** the time series of $y_n$, **c** the time series of $z_n$, **d** the time series of the cycling chaos signal

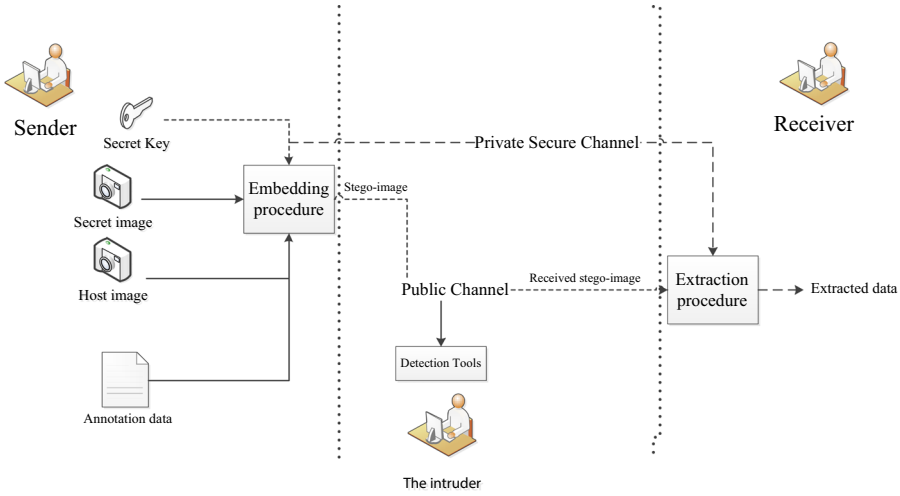


**Fig. 2** The schematic model of the proposed algorithm

these types is designed for a specific purpose. The private channel, which is assumed to be protected from eavesdrop, is only prepared for the sender and receiver. Therefore, the channel is called the secure private channel. The public channel, on the other hand, is reachable for every entity. Accordingly, any data that goes through this channel can be received by the intruder. While the private and public channels can work together, we assume that the private channel operates before the public channel starts to operate. We also assume that the data that is sent through the public channel is resistant against white noise, because the protocol of the transmission media can tackle such noise. The best example of a public channel is the Internet that can detect and correct the errors that occur during the transmission. It is done by TCP/IP protocols that are devised in each layer of the TCP/IP network [24].

Figure 2 shows the secret data transmission (concealment) process and the three different entities discussed before, where each entity has a different role. The sender sends stego-image with or without the annotation data. The receiver receives and extracts the transmitted stego-image using the key that passes through

the private channel. The intruder reads all the data transmitted by the public channel. In general, there are two kinds of intruders: the passive and the active intruders. Passive intruders adopts a specific algorithm to find out whether or not the transmitted data are carrier of the secret data. The active intruder, by contrast, reads, changes and resends the data to the receiver to deceive the receiver.

Although the proposed model is similar to conventional models (in their data hiding algorithm), it differs from them in some aspects. First, in the proposed model, we assume that the intruder does not have access to the private channel, and the private channel differs from the public channel physically or by means of time. On the contrary, in some existing models, it is assumed that the private channel is similar to the public channel in all aspects except that the data that are sent are encrypted before being transmitted.

Second, similar to the famous key-based security systems like sitekey and private key, the proposed model uses a secure secret key-based communication system in which the secret key is sent through a separate secure data channel. Unlike this, the conventional models embed the secret key (password) in the stego-image, and the receiver needs to know about the password before the extraction process.

Third, in the proposed model, we do not need to encrypt the secret data before embedding them in the host image. This is because the proposed algorithm itself provides enough security. On the other hand, in some existing steganographic models [5,26,27], an encryption algorithm like the AES algorithm should be applied on the secret image before the embedding process.

Fourth, the proposed model does not impose any limitation on the data embedded in the host image. In other words, the proposed model is capable of concealing every type of data such as images, texts and videos in the host image. In some existing models [10,12,28,29], on the other hand, we can only hide image type data in the host image.

Fifth, the proposed model is immune against the man-in-the-middle attack, as the sender and receiver pass the secret key through the private channel, and thus, the intruder cannot impersonate any of them.

Sixth, in the proposed model, the receiver only needs to know the stego-image for the extracting procedure. In some other models, the receiver should know both the stego-image and the host image to extract the sensitive data.

Finally, in some models, it is assumed that the intruder does not know the steganographic algorithm. However, according to the "Kerckhoffs principle" and the principles of modern information security, an information security algorithm must be available to everyone. Consequently, it is assumed that the proposed model is publicly known, and the intruder could know the algorithm we use for the data hiding.

## 4 Proposed algorithm

In this section, a new steganographic algorithm called CCSA is presented for concealing both the color and grayscale images. Characterized by cycling chaos systems along with PRNG, the proposed approach is applied to two different types of image hiding. The first type is the color image hiding that embeds a color secret image into a color host image. The second type is the grayscale image hiding that embeds a grayscale secret image into a host image. For the first type of image hiding, the secret image should be quantized; it is because the secret image is large and cannot be embedded directly to the host image. For the second type of hiding, on the other hand, the secret grayscale image can directly be embedded into the host image. Except for the quantization procedure, the algorithm uses the same methods for both types of hiding. In the following, we first focus on the key space of the proposed algorithm and then take a look at the embedding and extracting procedures.

### 4.1 The key space

In the proposed algorithm, the initial conditions ($x_0$, $y_0$, $z_0$) along with the control parameters of the cycling chaos function ($\lambda_1$, $\lambda_2$, $\lambda_3$, $m$ and $\gamma$) and the number of intervals $I$ serve as the key (K) of the transmission. Since $x_0$ ,$y_0$ and $z_0$ can be any double value within the range of $[-1.6, 1.6]$, $m$ can be in the range of $(0, 0.5]$ and $\gamma$ in the range of $[2.7, 3.0]$, the size of the key space is very large. Furthermore, $I$ as the integer part of the key can be any integer value in the range of $[1, 2^{30}]$. The structure of the key space is represented in Fig. 3.

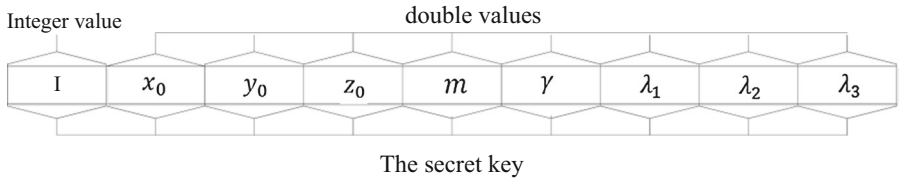


**Fig. 3** The key space of the proposed algorithm

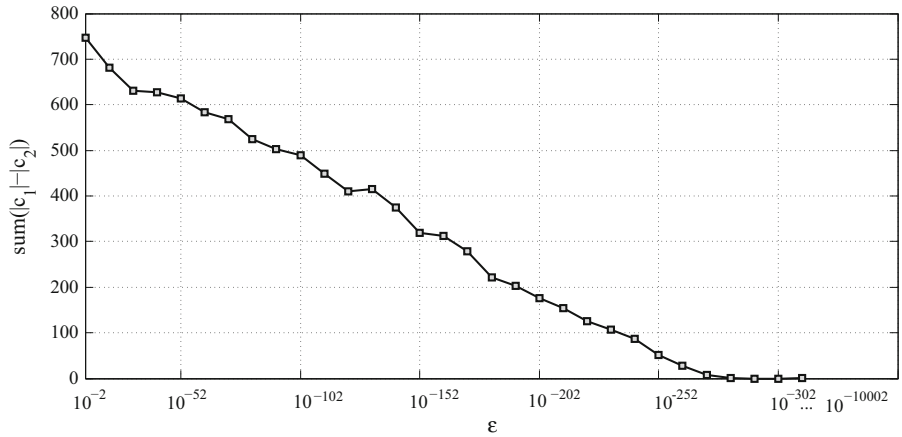


**Fig. 4** Sensitivity of the cycling chaos function to the changes in $\delta_x$

In the next, we discuss the sensitivity of the cycling chaos to the alteration of its parameters with concentrating on the maximum precision of the cycling chaos and showing the infeasibility of any brute force attack.

### 4.2 Finding the precision of the cycling chaos function

The chaos function is highly sensitive to the change of its initial conditions and its critical parameters, so even a slight change in each of them induces dramatic changes in the resultant chaos trajectory. Theoretically, the sensitivity of the cycling chaos to the alteration of its parameters is infinite. However, we can practically find the precision of the cycling chaos system where there is no strong reaction to the change of the parameters using Multiprecision Computing Toolbox [30] that is provided in MATLAB. Thanks to this toolbox, we can calculate the precision of the cycling chaos with thousands of decimal digits.

In order to measure the resistance of cycling chaos system to changes in the initial condition and control parameters, we first add a very small value, called $\delta$, to one of the parameters and leave the others intact. Then, we measure the sum of absolute difference (SAD) before and after the change. If SAD is large, $\delta$ is divided by 10. This process continues until the SAD is less than 1. We define the position that the sum of absolute difference is smaller than 1 as the precision of the cycling chaos.

To do so, we compare two different configurations of the cycling chaos where for the first one (called '$c_1$'), the parameters are set to $x_0 = -0.01$, $y_0 = 0.03$, $z_0 = 0.02$, $\lambda_1 = 3.0$, $\lambda_2 = 2.98$, $\lambda_3 = 2.87$, $\gamma = 3.05$ and $m = 0.25$ and for the second (called '$c_2$'), the parameters are set to the same values as the first one except $x_0$ that is set to $x_0 + \delta_x$. We set $\delta_x$ to $10^{-2}, 10^{-12}, 10^{-22} \ldots, 10^{-10002}$.

To compare these configurations, for each configuration, we first iterate the cycling chaos for 1,000 iterations and then compute the SAD between the two signals, which is found as,

$$\text{SAD} = \sum_{i=1}^{1000} |c_1^i - c_2^i|, \tag{1}$$

where $c_1^i$ and $c_2^i$ are the results of the $i$th iteration of $c_1$ and $c_2$ of the cycling chaos, respectively.

Figure 4 shows the SAD of the cycling chaos where $c_1$ and $c_2$ are used.

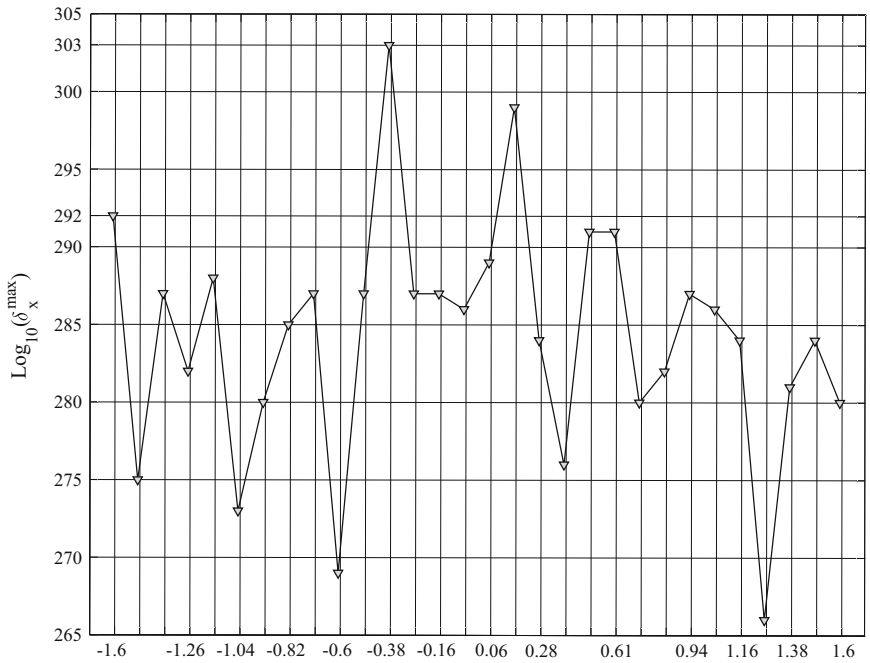


**Fig. 5** The precision of the cycling chaos against the domain of $x_0$

As shown in Fig. 4, there is a significant difference between $c_1$ and $c_2$ where $\delta_x$ is less than $10^{-282}$. For example, when $\delta_x = 10^{-202}$, the SAD of $c_1$ and $c_2$ is 190. Interestingly, within the area we have studied, the cycling chaos is always sensitive to changes; even when $\delta_x$ is $10^{-10002}$, there is a slight difference ($1.3 \times 10^{-10723}$) between configurations. However, the question is what is the response of the cycling chaos function when other initial values of $x_0$ are used. Figure 5 shows the maximum precision $\delta_x^{\max}$ of the cycling chaos for different values of $x_0$ where $x_0$ is in the range of $[-1.6, 1.6]$.

As shown in Fig. 5, the highest precision of the cycling chaos $\delta_x^{\max}$ for different values of $x_0$ is between $10^{-265}$ and $10^{-303}$. This means that the maximum precision of the cycling chaos is always high, and the cycling chaos is highly sensitive to the change of $x_0$, regardless of the value of $x_0$. We have also studied the other control parameters ($y_0, z_0, \lambda_1, \lambda_2, \lambda_3, \gamma, m$) and have seen the same results (due to space limitations, the results are not included in the paper).

The horizontal axis in Fig. 5 is on a logarithmic scale and the data fit a straight line. This suggests that the precision of the cycling chaos system decays logarithmically with $\delta_x$.

### 4.3 The sensitivity of the proposed algorithm to the change in the secret key

In this section, we examine the sensitivity of the proposed algorithm to alterations in its secret key. In order to measure the sensitivity of the proposed algorithm, we use the number of pixels change rate (NPCR), which measures the difference between images [31,32] and is defined as,

$$\mathrm{NPCR} = \frac{\sum_{i=0}^{M}\sum_{j=0}^{N} D_{i,j}}{M \times N} \times 100\,\%. \tag{2}$$

where $M$ and $N$ are height and width of the image and $D_{i,j}$ is the number of pixel differences, which is calculated as,

$$D_{i,j} = \begin{cases} 1, & \text{if } T_{i,j} \neq S_{i,j} \\ 0, & \text{otherwise} \end{cases}$$

where $T$ and $S$ are the stego-image and retrieved stego-image, respectively.

To calculate NPCR, first a secret image is embedded into the host image using the proposed algorithm with the secret key, which is set to $x_0 = -0.01$, $y_0 = 0.03$, $z_0 = 0.02$, $\lambda_1 = 3.0$, $\lambda_2 = 2.98$, $\lambda_3 = 2.87$, $\gamma = 3.05$, $m = 0.25$ and $I = 65535$. Then, the image is extracted with both the correct and wrong keys. The

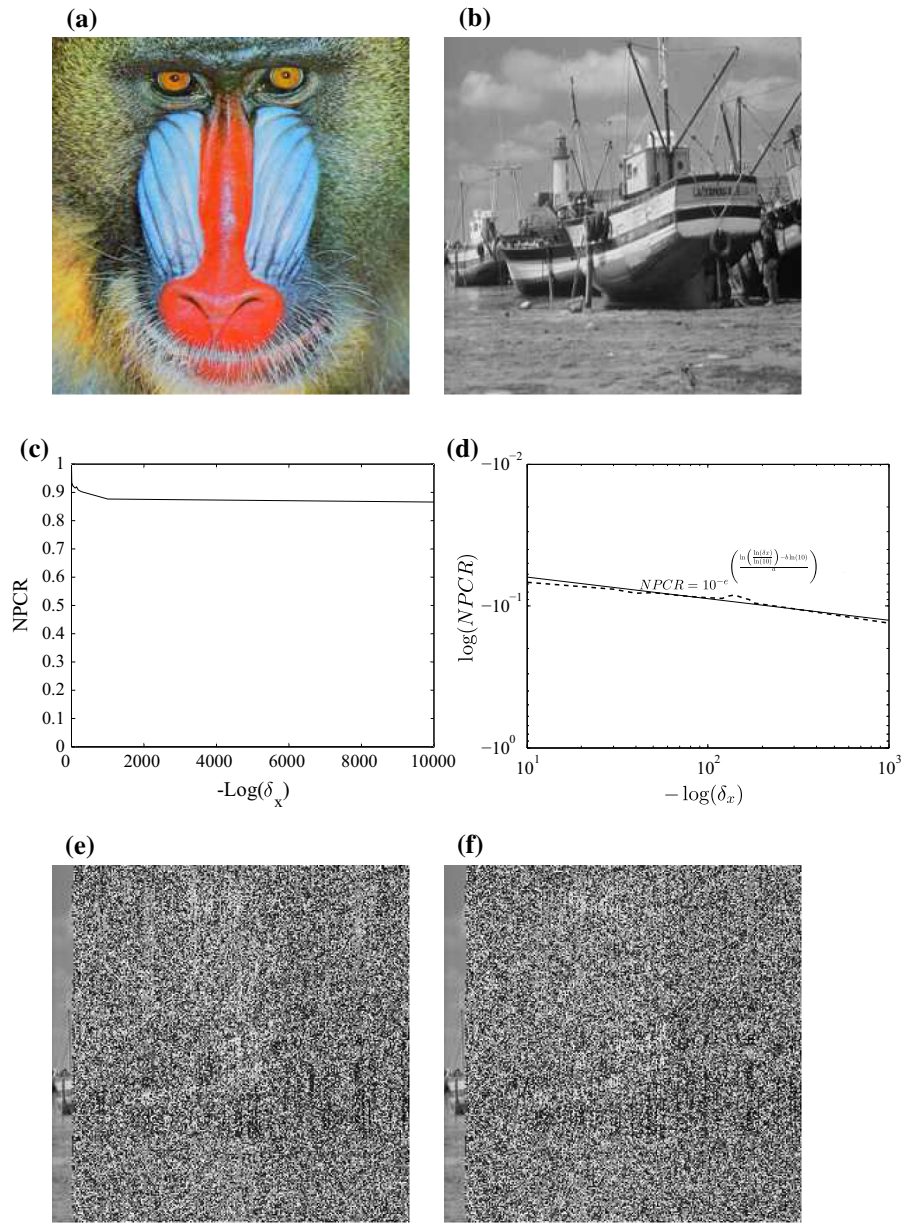


**Fig. 6** The NPCR obtained by the proposed algorithm for different values of $\delta_x$. **a** The host image "Baboon," **b** the secret image "Boat," **c** the NPCR versus $-\log(\delta_x)$, **d** the logarithm of the NPCR versus $-\log(\delta_x)$ where the plot is scaled logarithmically, **e** the retrieved secret image with the wrong secret key where $\delta_x = 1900$, **f** the retrieved secret image with the wrong secret key where $\delta_x = 10000$

wrong key component values are the same as those of the correct key except $x_0$ that is set to $x_0 + \delta_x$. $\delta_x$ is set to $10^{-10}, 10^{-20}, \ldots, 10^{-10000}$.

Figure 6 shows $\delta_x$ values versus the NPCR values obtained by the proposed algorithm where the grayscale image is "Boat" and the color image is "Baboon" both with $256 \times 256$ pixels. The grayscale image is used as the secret, and the color image is used as host images.

Figure 6 shows the logarithm of the NPCR versus $-\log(\delta_x)$ on a log–log scale, that is, the $y$ axis is scaled logarithmically twice and the $x$ axis is scaled logarithmically. Therefore, the data suggest that NPCR decays as,

$$\log(\log(\text{NPCR})) = a \log(-\log(\delta_x)) + b, \tag{3}$$

thus

$$\mathrm{NPCR} = 10^{-e^{\left(\frac{\ln\left(\frac{\ln(\delta x)}{\ln(10)}\right)-b\ln(10)}{a}\right)}} \quad (4)$$

where $a$ is the gradient and $b$ is the intersection of the line in Fig. 6. Since $a$ is a negative number, the graph suggests that even a very small $\delta_x$ ($10^{-1800}$) results in a large NPCR. This means that applying an infinitesimally small change to the key results in a different image; therefore, the different number of configurations for the key is very large (in the order of $10^{2000}$).

Consequently, we can claim that the brute force attack or any other attack that requires checking all the possible values is practically impossible since the cycling chaos that is the backbone of the proposed algorithm is immensely sensitive to the alteration of its parameters.

### 4.4 The embedding procedure

The proposed algorithm is designed to embed the secret data in a host image. Although the proposed algorithm (CCSA) is able to hide any form of data such as text, video or image into the host image, in this paper we only focus on image hiding. The secret image can be a grayscale or 256-color palette-based image. The embedding procedure mainly consists of two main parts: the cycling chaos and the PRNG. The cycling chaos is used for generating the seed values of PRNG. PRNG is utilized for generating pseudorandom sequences of pixel positions and channels of the pixels. This means that it determines in what pixels of the host image and in what channel (R, G or B) of the pixel the secret image is embedded. The block diagram of the embedding procedure of the proposed algorithm for the first type of hiding is represented in Fig. 7. The block diagram and the pseudo-code of the second type of hiding is omitted due to its similarity to the first type. As mentioned before, the only difference between these two types is that in the first type, the color quantization process is used, while in the second type, the secret image is directly embedded into the host image.

To clarify the embedding procedure, its pseudo-code is presented as below,

The embedding procedure
1. set $x_0, y_0, z_0, \lambda_1, \lambda_2, \lambda_3, \gamma, I, m, S, H, A$
2. divide the interval $[-2, 2]$ named $d$ into $I$ equal subintervals
3. set $T$ with $H$
4. $[c_t, q_S] = \mathrm{colquan}(S)$
   $x_n = x_0, y_n = y_0, z_n = z_0$
5. $b_d = \mathrm{binconv}(q_S, c_t, A)$
6. $s_e = \mathrm{cyclingchaos}(x_n, y_n, z_n, \lambda_1, \lambda_2, \lambda_3, \gamma, m)$
7. generate a pseudorandom class called $R$ with the seed value equal to $s_e$
8. $r_1 = randsqgen(R, n_c)$
9. for $i = 1{:}n_c$
10. $i_c = r_1^i$
11. $s_e = \mathrm{cyclingchaos}(x_n, y_n, z_n, \lambda_1, \lambda_2, \lambda_3, \gamma, m)$
12. generate a pseudorandom class called $R$ with the seed value equal to $s_e$
13. $r_2 = randsqgen(R, n_r)$
14. for $(j = n_r)$
15. $i_r = r_2^j$
16. $s_e = \mathrm{cyclingchaos}(x_n, y_n, z_n, \lambda_1, \lambda_2, \lambda_3, \gamma, m)$
17. generate a pseudorandom class called $R$ with the seed valne equal to $s_e$
18. $r_3 = randsqgen(R, n_o)$
19. for $k = 1:3$
20. $i_o = r_3^k$
21. extract three bits from the $b_d$ and put it in $S_b$
22. $\acute{H} = H_{i_c,i_r,i_o} + Sb - (H_{i_c,i_r,i_o} mod 8)$
23. if $(\acute{H} < H(i_c, i_r, i_o))$
24. $g = \acute{H} + 8$
25. if $g \le 255$ and $|H(i_c, i_r, i_o) - g| \le |H(i_c, i_r, i_o) - \acute{H}|$
26. $\acute{H} = g$
27. if $\acute{H} > H$
28. $g = \acute{H} - 8$
29. if $g \ge 0$ and $|H(i_c, i_r, i_o) - g| < |H(i_c, i_r, i_o) - \acute{H}|$
30. $T(i_c, i_r, i_o) = g$

end for
end for
end for
end Procedure

A more detailed description of the proposed algorithm is as follows:

Step 1: At the beginning, the sender needs to set the initial values for the parameters of the embedding procedure, including the initial conditions $(x_0, y_0, z_0)$ and the control parameters $(\lambda_1, \lambda_2, \lambda_3, \gamma, m)$ of the cycling chaos. The number of intervals $I$ is another

important component of the secret key that should be defined by the sender. The sender also needs to provide the host image ($H$), the secret image ($S$) and the annotation data($A$) before the hiding procedure. As mentioned, the secret key (K) should be sent through a secure private channel to the receiver (see Fig. 2).

Step 2: Before the embedding process, we also need to discretize the interval $d$ into $I$ equal subintervals with the length of $\frac{4}{I}$. The index of a subinterval is used for determining the seed of PRNG.

Step 3: Before embedding the secret image into the host image, we initialize the stego-image $T$ with the host image $H$.

Step 4: In this step, in order to reduce the number of colors in the secret image, a color quantization procedure in the form of the minimum variance [22] is applied to the secret color image. This process returns two matrices called $q_S$, which is a quantized color secret image, and $c_t$, which is a color map for the quantized color secret image. The length of $q_S$ is $n_r \times n_c$ where $n_c$ is the number of columns and $n_r$ is the number of the rows in the host image. The length of $c_t$ is $256 \times 3$ where "256" is the number of gray levels in the quantized-host image. Note that for the second type of image hiding this step is omitted.

Step 5: In this step, each of $A$, $q_S$ and $c_t$ are first combined together, forming a new one-dimensional array. Then, the combined data array is converted to a one-dimensional binary data array called $b_d$.
In order to determine in what pixels of the host image $H$ and in what channel of the pixels the secret binary data ($b_d$) are embedded, we use the cycling chaos in conjunction with PRNG. To do so, in step 5–9, the selection order of columns of $H$ is determined. Similarly, in steps 10–14, the selection order of rows and in steps 15–19, the selection order of the channel of the pixels are determined.

Step 6: The integral part of the embedding procedure is the cycling chaos function, and hence, we describe it in more details in the following:

procedure cyclingchaos($x_n$, $y_n$, $z_n$, $\lambda_1$, $\lambda_2$, $\lambda_3$, $\gamma$, $m$)
begin
$x_{n+1} = \lambda_1 x_n - x_n^3 - \gamma |y_n|^m x_n$
$y_{n+1} = \lambda_2 y_n - y_n^3 - \gamma |z_n|^m y_n$
$z_{n+1} = \lambda_3 z_n - z_n^3 - \gamma |x_n|^m z_n$
$u = max(x_{n+1}, y_{n+1}, z_{n+1})$
for $i = 1{:}I$-1
if ($d^i \leq u$ and $d^{i+1} > u$)
$se = i$, break
end for
$x_n = x_{n+1}$, $y_n = y_{n+1}$, $z_n = z_{n+1}$
return $s_e$
end procedure

We first iterate the cycling chaos once and then select the highest value among $x_{n+1}$, $y_{n+1}$ and $z_{n+1}$ as $u$. Then, we search the interval that is discretized into $I$ equal subintervals in order to find the most appropriate subinterval. The first subinterval in the domain of the cycling chaos that is less than $u$ is selected as the selected subinterval. The corresponding index value of the selected interval is used as the seed of the PRNG in the following. Finally, the cycling chaos signal is updated.

Step 7: As mentioned before, PRNG is deterministic according to seed value $s_e$ and the exact sequence generated by PRNG can be obtained by using the same seed value. Therefore, if the receiver does not enter the exact value of the key ($K$), the seed value will be completely different, resulting in a completely different sequence.

Step 8: A pseudorandom permutation of the $x$-coordinates of the pixel positions of the host image, called $r_1$, is generated. Because the permutation is pseudorandom, by having the seed of the pseudorandom generator, the same permutation can be generated at any time.

Step 9: For all the columns of the host image, the steps 10–30 are taken.

Step 10: The $i$th element of the pseudorandom sequence $r_1$ is selected and stored in $i_r$.

Steps 11–13: In these steps, a pseudorandom permutation of $y$-coordinate of pixel positions of the host image is generated. These steps are similar to steps 6 to 8. The only difference here is that in step 11, the initial conditions of the cycling chaos ($x_n$, $x_y$ and $z_n$) are equal to the last values of $x_n$, $y_n$, $z_n$ obtained in step 7.

Step 14: For all the rows of the host image ($n_r$), the following steps are taken.

Step 15: The $j$th element of the pseudorandom sequence $r_2$ is selected and stored in $i_c$.

Steps 16-18: In these steps, a pseudorandom sequence of the channels of the ($i_r$, $i_c$)-pixel is generated.

Step 19: For all the channels of the $(i_r, i_c)$-pixel, the following steps are taken.

Step 20: In order to set $b_d$ into $i_o$- channel of $(i_r, i_c)$-pixel of the host image $H$, we extract the secret data from $b_d$, 3-bits by 3-bits and put them into $S_b$. $S_b$ is a variable between 0 and 7.

Steps 21–30: In steps 21–29, the 3-bit extracted binary data ($S_b$) are embedded in the $i_o$th color plane ($1 \leq i_o \leq 3$) of $i_c$th column of $i_r$th row of the host image $H$. Inspired by the OPVSP procedure used in Yu et al.'s method [5], the color substitution process embeds $S_b$ in the LSB plane of $H(i_c, i_r, i_o)$ with the least degradation. In steps 21–29, the algorithm finds the nearest neighbor color to the color of the image pixel in which $S_b$ is embedded. In step 21, the differential value of LSB plane and $S_b$ is added with the value of $H(i_c, i_r, i_o)$. In steps 22 to 29, the optimal color value $\acute{H}$ that is the nearest possible value to $H(i_c, i_r, i_o)$ is calculated and stored in $T(i_c, i_r, i_o)$.

### 4.5 The extracting procedure

For the extraction procedure, the receiver must know the secret key $K$ before the extraction process is performed. Note that the size of the secret image is hidden in the host image as annotation data, and the receiver can easily extract it by using the secret key. The extraction procedure is performed like the inverse of the embedding procedure. More specifically, first the secret key $K$ and the stego-image $T$ are used as an input of the extraction procedure. Then, when $K$ is determined, by utilizing the cycling chaos function in conjunction with PRNG, the annotation data and the quantized-secret image $S$ are sequentially extracted from the stego-image. The block diagram of the extraction procedure of the proposed algorithm is represented in Fig. 7. The pseudo-code of the extraction procedure is presented later in this paper (Fig. 8).

The extraction procedure

1. get $x_0, y_0, z_0, \lambda_1, \lambda_2, \lambda_3, \gamma, I, m, T$ from sender
2. divide the interval $[-2, 2]$ named $d$ into $I$ equal sub-intervals
   $x_n = x_0, y_n = y_0, z_n = z_0$
3. $s_e = \text{cyclingchaos}(x_n, y_n, z_n, \lambda_1, \lambda_2, \lambda_3, \gamma, m)$
4. generate a pseudorandom class called $R$ with the seed value equal to $s_e$
5. $r_1 = randsqgen(R, n_c)$

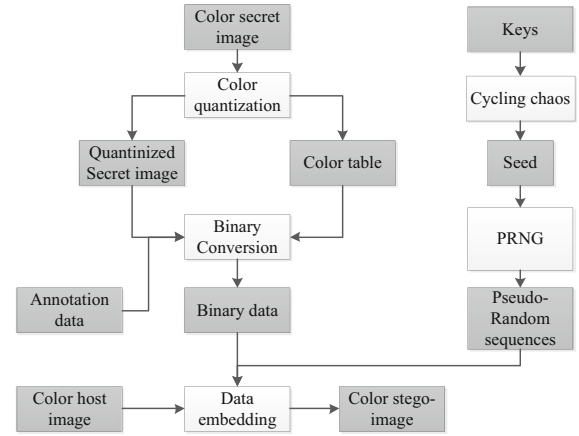


**Fig. 7** The block diagram of the embedding procedure for the first type of hiding

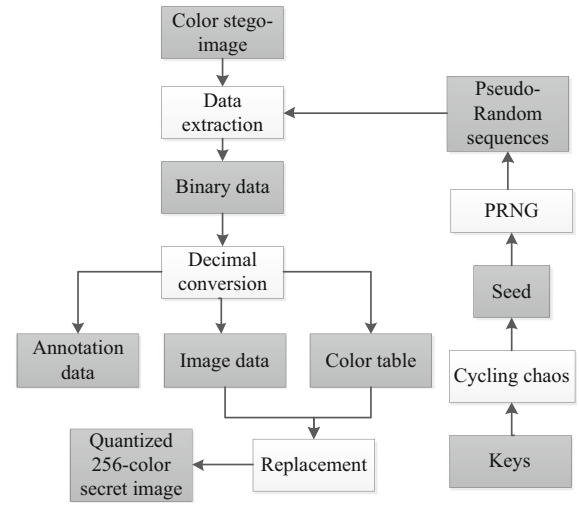


**Fig. 8** The block diagram of the extraction procedure

6. for i = 1:$n_c$
7. $i_c = r_1^i$
8. $s_e = \text{cyclingchaos}(x_n, y_n, z_n, \lambda_1, \lambda_2, \lambda_3, \gamma, m)$
9. generate a pseudorandom class, $R$ with the seed value equal to $s_e$
10. $r_2 = randsqgen(R, n_r)$
11. for $(j = n_r)$
12. $i_r = r_2^j$
13. $s_e = \text{cyclingchaos}(x_n, y_n, z_n, \lambda_1, \lambda_2, \lambda_3, \gamma, m)$
14. generate a pseudorandom class called $R$ with the seed value equal to $s_e$
15. $r_3 = randsqgen(R, n_o)$
16. for k = 1:3
17. $i_o = r_3^k$

18. $S_b = T(i_c, i_r, i_o) mod 8$
19. Insert $S_b$ to $b_d$ sequentially
end for
end for
end for
20 $[q_S, c_t, \mathrm{A}] = \mathrm{IntConv}(b_d)$
21 $S = \mathrm{Replacement}(c_t, q_S)$
end Procedure

Because most steps of the extraction procedure are the same as those of the embedding procedure, we only describe the different steps.

Step 1: In this step, the receiver uses the secret key $K$ that is received through the private channel as the input of the extraction procedure. The receiver also gets the stego-image $T$ through the public channel from the sender.

Step 18: After the $x$-coordinate $i_c$ and $y$-coordinate $i_r$ and the channel (RGB) of the pixel in the stego-image $T$ in which the secret binary data are hidden are determined, the algorithm extracts the secret binary data 3-bits by 3-bits from the LSBs of $i_o$th channel of $(i_r, i_c)$th pixel of $T$ and inserts them into a binary vector called $b_d$.

Step 19: In this step, $b_d$ is converted into an integer vector $A_l$ and then is converted to three vectors including the 8-bit index data $q_S$, the color table (the palette) $c_t$ and the annotation data $A$. Note that the size of the secret image is stored in $A$, so $A$ must first be extracted from $A_l$ and then $q_S$ and $c_t$ can be extracted.

Step 20: In the replacement procedure, the quantized 8-bit secret image with its color map is replaced with secret image $S$.

## 5 Experimental results

In this section, we first test and compare the quality of the proposed algorithm with Lin et al.s [12] and Yu et al.'s hiding schemes [5] on two different sets of experiments. Then, we study the performance of the proposed algorithm, concentrating on visual, RS, byte and Chi-square attacks.

The proposed algorithm has nine parameters that are used as the key of the transmission. The parameters are initial conditions and control parameters of the cycling chaos function and the number of intervals $I$. By default, we set the cycling chaos parameters to $x_0 = -0.01$, $y_0 = 0.03$, $z_0 = 0.02$, $\lambda_1 = 3.0$, $\lambda_2 = 2.98$, $\lambda_3 = 2.87$, $\gamma = 3.05$, $m = 0.25$ and $I = 65535$.

The first set of experiments is used for the first type of hiding (hiding a color secret image in a true color image) and the second set of experiments for the second type of hiding (hiding a grayscale image in a true color image). We use the same test images as those used in [5,12].

In the first set of experiments, we use six color images with $512 \times 512$ pixels. These images are "Airplane," "Baboon," "House," "Lena," "Peppers" and "Sailboat" [33] that serve as both the host and secret images. As mentioned before, in order to make the size of the secret image suitable to be embedded in the host image, we use a color quantization method in the form of minimum variance [22]. The color quantization turns each 24-bit secret image into a 256-color palette-based image. The quantized-secret image is then embedded in the host image.

In order to compare the quality of the proposed algorithm with Lin et al.'s and Yu et al.'s hiding schemes, we use the peak signal-to- noise ratio (PSNR) measurement technique [5,12,34,35]. The PSNR for grayscale images is defined as,

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right), \tag{5}$$

where

$$\mathrm{MSE} = \frac{1}{n_r \times n_c} \sum_{i=1}^{n_r} \sum_{j=1}^{n_c} \left(H_{ij} - T_{ij}\right)^2, \tag{6}$$

where MSE is the mean square error, $H$ and $T$ are the host and stego-images.

Similarly, for color images the PSNR is defined as,

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{255^2}{\mathrm{MSE}_{avg}}\right), \tag{7}$$

where $\mathrm{MSE}_{\mathrm{avg}}$ is the average of mean square error of three different colors of the image.

Note that PSNR measures the quality of the image, that is, the higher PSNR value, the higher the quality of the resulting image. Table 1 compares the PSNR values of stego-images obtained from the proposed algorithm, Lin et al. [12] and Yu et al.'s methods [5].

**Table 1** Comparison between the proposed method and the schemes in [12] and [5] of the PSNR values of the stego-images from the fist type of hiding

| Host image | Airplane | | | Baboon | | | House | | |
|---|---|---|---|---|---|---|---|---|---|
| Secret image | [5] | [12] | CCSA | [5] | [12] | CCSA | [5] | [12] | CCSA |
| Airplane | 41.835 | 39.395 | 44.241 | 41.870 | 39.162 | 44.221 | 41.833 | 39.195 | 44.213 |
| Baboon | 41.863 | 39.247 | 44.254 | 41.860 | 29.174 | 44.259 | 41.860 | 39.168 | 44.243 |
| House | 41.879 | 39.271 | 44.287 | 41.873 | 39.121 | 44.268 | 41.874 | 39.125 | 44.269 |
| Lena | 41.863 | 39.304 | 44.230 | 41.872 | 39.168 | 44.212 | 41.879 | 39.177 | 41.890 |
| Peppers | 41.856 | 39.358 | 44.227 | 41.878 | 39.090 | 44.224 | 41.881 | 39.131 | 44.217 |
| Sailboat | 41.871 | 39.315 | 44.234 | 41.873 | 39.189 | 44.224 | 41.864 | 39.232 | 44.237 |
| Host image | Lena | | | Peppers | | | Sailboat | | |
| Secret image | [5] | [12] | CCSA | [5] | [12] | CCSA | [5] | [12] | CCSA |
| Airplane | 41.868 | 39.170 | 44.210 | 41.582 | 39.123 | 44.399 | 41.877 | 39.194 | 44.223 |
| Baboon | 41.872 | 39.171 | 44.258 | 41.615 | 39.191 | 44.437 | 41.868 | 39.204 | 44.241 |
| House | 41.879 | 39.091 | 44.248 | 41.598 | 39.143 | 44.436 | 41.886 | 39.152 | 44.262 |
| Lena | 44.213 | 39.125 | 41.624 | 44.203 | 39.178 | 44.395 | 41.866 | 39.161 | 44.222 |
| Peppers | 41.873 | 39.099 | 44.204 | 41.607 | 44.404 | 39.068 | 41.857 | 39.124 | 44.226 |
| Sailboat | 41.871 | 39.178 | 44.223 | 41.612 | 39.196 | 44.422 | 41.872 | 39.219 | 44.241 |

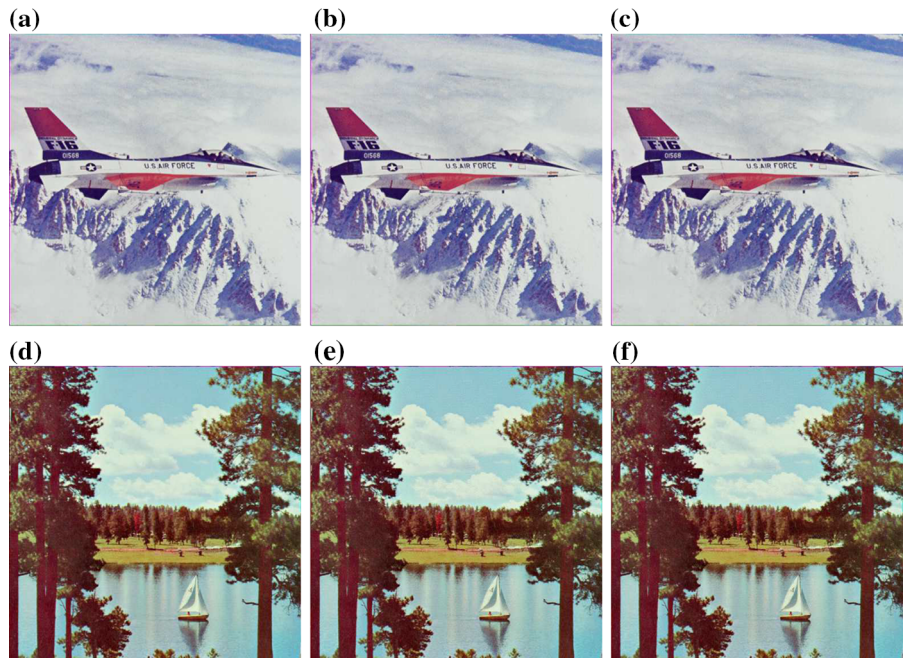


**Fig. 9** Some of the retrieved secret images where the host image is "Baboon." **a** The secret image Airplane, **b** the quantized-secret image Airplane, **c** the retrieved secret image Airplane, **d** the secret image Sailboat, **e** the quantized secret image Sailboat, **f** the retrieved secret image Sailboat

As shown in Table 1, the PSNR value of stego-images obtained by the proposed algorithm is much higher than that of stego-images obtained by the other algorithms. Figure 9 shows some of the retrieved secret images obtained by the proposed algorithm from the first type of hiding. In Fig. 9a, c, the original secret

image Airplane and Sailboat, in Fig. 9b, d, the quantized versions and in Fig. 9e, f the retrieved versions are represented.

As shown in Fig. 9, the images are retrieved successfully from the host image, and they are fairly accurately similar to their corresponding original images. Comparing the Fig. 9b, d indicates that the images are successfully quantized to 256 colors, and there is no noticeable difference between the original 32-bit color images with their respective 8-bit quantized images. Furthermore, as shown in Fig. 9c, e, the secret images, after being retrieved from the decoding process, are quiet similar to their respective original images.

In order to show the quality of the stego-images, we present Fig. 10. In Fig. 10a, c, e, three original images are represented and in Fig. 10b, d, f, the images are shown after being embedded by the secret image Airplane.

As shown in Fig. 10, there is no remarkable difference between the images in the first column and the second column; thus, we can infer that the proposed algorithm promisingly conceals the secret image. In the second type of image hiding, we directly embed a $512 \times 512$ grayscale image into a color image with the same size.

For the second type of image hiding, we use seven grayscale secret images: "Airplane," "Baboon," "Barbara," "Boat," "Lena," "Peppers" and "Sailboat." For host images, we use the same color images as those employed for the first type of hiding. Table 2 shows the PSNR values of the stego-images obtained by the proposed algorithm as well as the other two competitive algorithms over the second set of experiment.

Because Lin et al.s scheme did show promising results in PSNR-based comparisons presented earlier in this paper, in the following we only compare the proposed algorithm with Yu et al.s scheme.

### 5.1 Quality index

To statistically measure the quality of stego-images, we use quality index [5,36] that is defined as,

$$Q = \frac{4\sigma_{HT}\bar{H}\bar{T}}{\left(\sigma_H^2 + \sigma_H^2\right)\left[\left(\bar{H}\right)^2 + \left(\bar{T}\right)^2\right]}, \tag{8}$$

where,

$$\bar{H} = \frac{1}{n}\sum_{i=1}^{n} H_i, \qquad \bar{T} = \frac{1}{n}\sum_{i=1}^{n} T_i,$$

$$\sigma_H^2 = \left(\frac{1}{n-1}\right)\sum_{i=1}^{n}\left(H_i - \bar{H}\right)^2,$$

$$\sigma_T^2 = \left(\frac{1}{n-1}\right)\sum_{i=1}^{n}\left(T_i - \bar{T}\right)^2.$$

In the above, $n$ is the number of pixels in the image, $H$ is the host image and $T$ is the stego-image. The value of $Q$ is between $-1$ and 1 where $-1$ means that there is no similarity between two images and 1 means that they are identical. Table 3 represents the quality index of stego-images for the first type of hiding.

As shown in Table 3, in most cases, the quality of the stego-images obtained by the proposed algorithm is much better than that of the stego-images obtained by Yu et al.'s scheme. Even though Yu et al.'s method offers better results on "Baboon" image, for other five secret images, the quality index gained by the proposed algorithm is much better.

### 5.2 Hiding capacity

One important criterion in steganographic algorithms is the hiding capacity of sensitive information. The more data the algorithm can embed, the more powerful the algorithm. Since the proposed algorithm can embed sensitive data in three LSBs of all channels of all image pixels, its hiding capacity is 3/8. Table 4 shows the comparison between the proposed algorithm and the LSBs-based algorithms proposed in [5,34,35,37] in terms of hiding capacity.

As shown in Table 4, the proposed algorithm has the highest hiding capacity. Since the proposed algorithm does not impose any restriction for setting secret information into the host image, we can potentially fill all the bits of the host image with the secret data. However, to maintain the quality of the stego-images, we only set information into the 3 rightmost bits of all the channels of the host image.

### 5.3 Visual attacks

In this section, the stego-images obtained by the proposed algorithm are exposed to visual attacks. Visual attack is defined as visually examining the stego-images in order to find a difference between the stego-images and their corresponding host images. Figures 10 and 9 show the stego-images obtained by the proposed algorithm where the secret image is "Airplane" and the host images are "Lena," "Baboon" and "House."

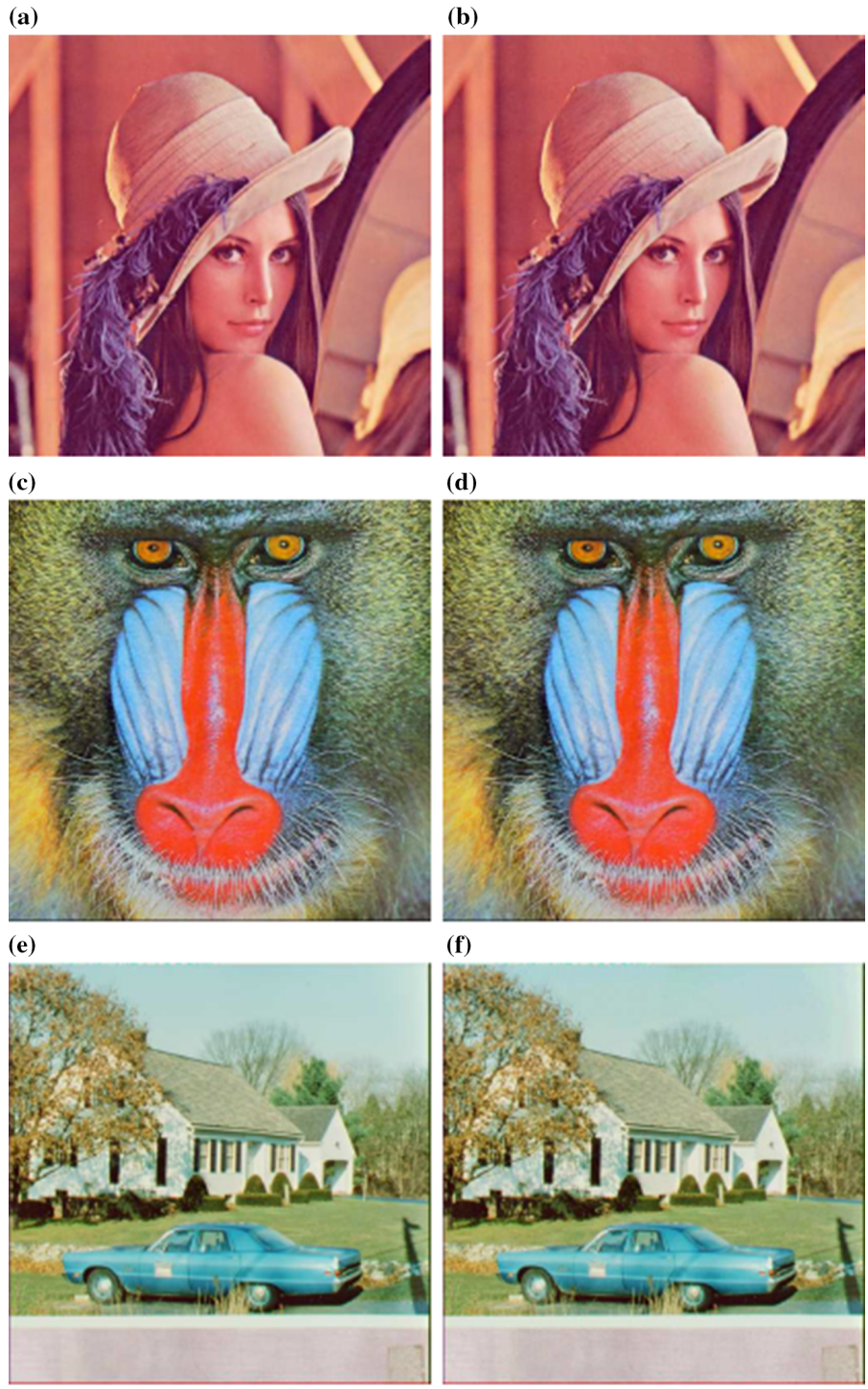


**Fig. 10** Some of the stego-images from the experimental results of the first type of hiding, where the secret image Airplane is embedded. **a** The original host image Lena, **b** the stego-image Lena, **c** the original host image Baboon, **d** the stego-image Baboon, **e** the original host image House, **f** the stego-image House

As seen in Fig. 10, the quality of stego-images is very high and an attacker cannot visually differentiate between stego-images and their corresponding original images.

The histogram distribution of the images that measures the undetectability of secret information [3] is studied in this paper to show the changes applied to the images. If there is no significant difference between

**Table 2** Comparison between proposed method and the schemes in [12] and [5] of the PSNR values of the stego-images from the experiment gray scale

| Color image | Airplane | | | Baboon | | | House | | |
|---|---|---|---|---|---|---|---|---|---|
| Gray scale | [5] | [12] | CCSA | [5] | [12] | CCSA | [5] | [12] | CCSA |
| Airplane | 39.151 | 41.692 | 44.421 | 39.177 | 41.944 | 44.408 | 39.124 | 41.980 | 44.407 |
| Baboon | 39.165 | 41.923 | 44.301 | 39.187 | 41.943 | 44.286 | 39.129 | 41.910 | 44.262 |
| Barbara | 39.150 | 41.968 | 44.238 | 39.180 | 41.949 | 44.215 | 39.125 | 41.922 | 44.223 |
| Boat | 39.153 | 41.948 | 44.125 | 39.180 | 41.948 | 44.109 | 39.126 | 41.930 | 44.102 |
| Lena | 39.159 | 41.914 | 44.235 | 39.178 | 41.946 | 44.213 | 39.124 | 41.915 | 44.209 |
| Peppers | 39.164 | 41.975 | 44.221 | 39.185 | 41.942 | 44.195 | 39.122 | 41.922 | 44.210 |
| Sailboat | 39.169 | 41.978 | 44.295 | 39.185 | 41.951 | 44.265 | 39.128 | 41.946 | 44.268 |
| Color image | Lena | | | Peppers | | | Sailboat | | |
| Gray scale | [5] | [12] | CCSA | [5] | [12] | CCSA | [5] | [12] | CCSA |
| Airplane | 39.177 | 41.956 | 44.393 | 39.086 | 41.715 | 44.605 | 39.045 | 41.937 | 44.427 |
| Baboon | 39.189 | 41.940 | 44.270 | 39.095 | 41.678 | 44.453 | 39.141 | 41.945 | 44.287 |
| Barbara | 39.185 | 41.943 | 44.217 | 39.088 | 41.682 | 44.412 | 39.145 | 41.936 | 44.227 |
| Boat | 39.181 | 41.948 | 44.081 | 39.102 | 41.669 | 44.277 | 39.149 | 41.934 | 44.103 |
| Lena | 39.181 | 41.947 | 44.191 | 39.092 | 41.689 | 44.403 | 39.146 | 41.938 | 44.228 |
| Peppers | 39.180 | 41.927 | 44.196 | 39.095 | 41.671 | 44.372 | 39.144 | 41.951 | 44.204 |
| Sailboat | 39.175 | 41.944 | 44.267 | 39.103 | 41.670 | 44.455 | 39.155 | 41.928 | 44.290 |

**Table 3** Quality index of stego-images obtained by the proposed algorithm and Yu et al.'s scheme from the first type of hiding

| Host image | Airplane | | Baboon | | House | | Lena | | Peppers | | Sailboat | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Secret image | [5] | CCSA | [5] | CCSA | [5] | CCSA | [5] | CCSA | [5] | CCSA | [5] | CCSA |
| Airplane | 0.9990 | 0.9990 | 0.9984 | 0.9992 | 0.9988 | 0.9997 | 0.9979 | 0.9986 | 0.9983 | 0.9990 | 0.9992 | 0.9998 |
| Baboon | 0.9999 | 0.9991 | 0.9999 | 0.9996 | 0.9999 | 0.9996 | 0.9999 | 0.9985 | 0.9994 | 0.9990 | 0.9999 | 0.9998 |
| House | 0.9988 | 0.9992 | 0.9983 | 0.9987 | 0.9992 | 0.9996 | 0.9980 | 0.9985 | 0.9986 | 0.9990 | 0.9993 | 0.9998 |
| Lena | 0.9988 | 0.9991 | 0.9985 | 0.9988 | 0.9992 | 0.9996 | 0.9982 | 0.9986 | 0.9985 | 0.9990 | 0.9994 | 0.9998 |
| Peppers | 0.9987 | 0.9991 | 0.9985 | 0.9988 | 0.9992 | 0.9996 | 0.9982 | 0.9986 | 0.9986 | 0.9990 | 0.9994 | 0.9998 |
| Sailboat | 0.9988 | 0.9991 | 0.9984 | 0.9987 | 0.9992 | 0.9996 | 0.9980 | 0.9985 | 0.9984 | 0.9990 | 0.9994 | 0.9998 |

**Table 4** Comparison between the proposed method and the schemes proposed in [5,34,35,37] in terms of hiding capacity

| Method | Hiding capacity |
|---|---|
| Proposed method | 3/8 |
| Ghebleh et al.s hiding scheme [37] | 2.25/8 |
| Yu et al.s hiding scheme [5] | 1/3 |
| Rongrong et al.s hiding scheme [34] | 1/12 |
| Liu et al.s hiding scheme [35] | 3/128 |

the histogram distribution of the host image and its stego-image, the algorithm successfully passes this test. Figure 11 shows the histogram distribution of stego-image "House" in which the secret image "Lena" is embedded. As shown in Fig. 11, the histogram of the original image "House" is quite similar to that of stego-image house obtained by the proposed algorithm. This indicates that the proposed algorithm is resistant against histogram attack.

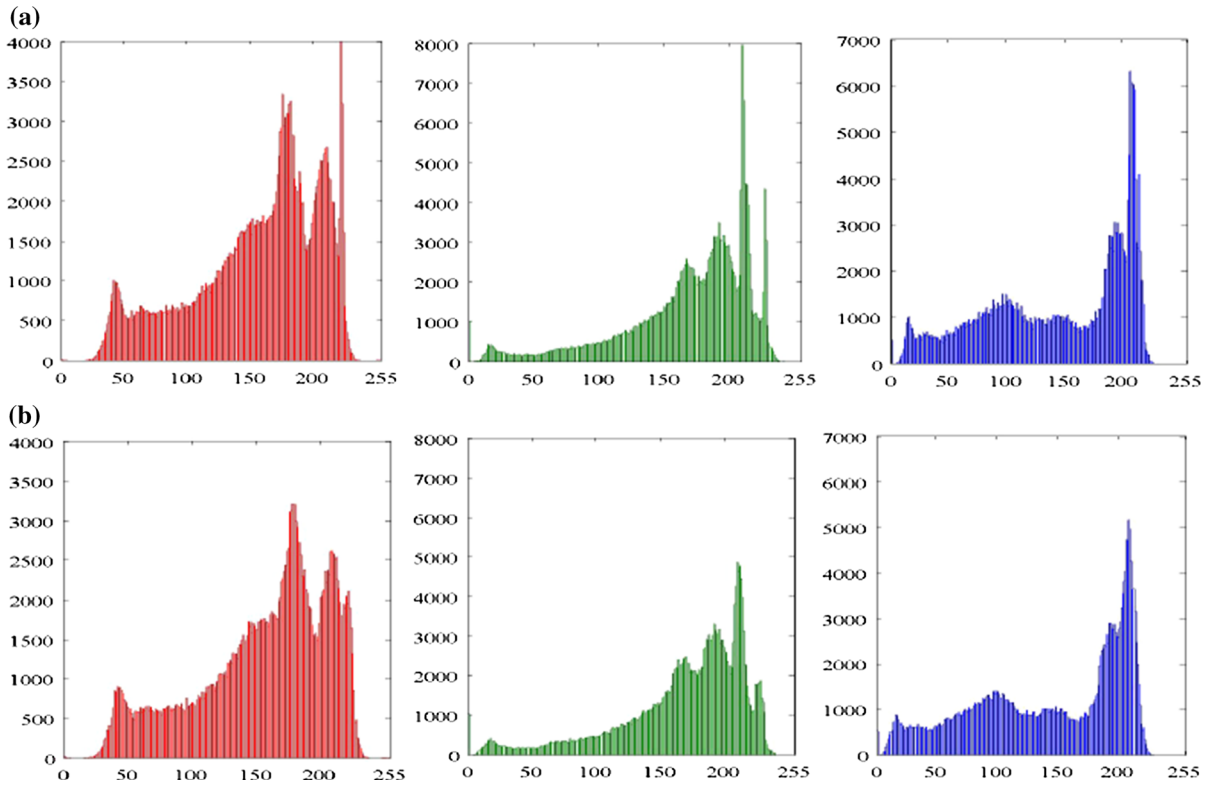


**Fig. 11** Histogram of the color image "House". **a** Histogram of original House image, **b** histogram of stego-image House obtained by the proposed algorithm

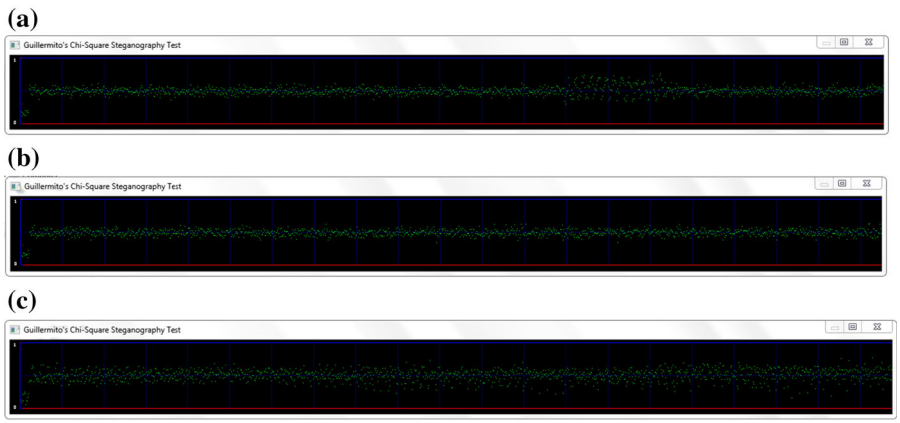


**Fig. 12** Chi-square for color image "House" where secret image Lena is embedded. **a** Chi-square for "House" host image, **b** Chi-square for stego-image House obtained by the proposed algorithm, **c** Chi-square for stego-image "House" obtained by Yu et al.'s method [5]

### *5.3.1 Chi-square test*

Because the proposed steganographic algorithm embeds the secret image in the LSB plane of the host image, the LSB plane is changed after the embedding procedure. In this section, we apply the Chi-square test [38] that is a statistical measure to find out if there is a significant difference between the expected frequency and the observed frequency. To apply the Chi-square test, we use Guillermito software [39] that is an open- source program. Figure 12 shows the Chi-square test obtained by the proposed algorithm and Yu et al.'s method for

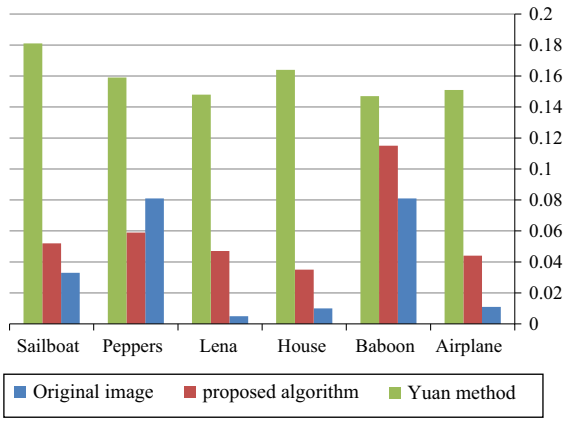


**Fig. 13** RS steganalysis comparison between the proposed algorithm and Yu et al.'s method [5]

the host image "House," where the secret image is a 512 × 512 "Lena" color image. In this figure, the red line shows the result of Chi-square test. If the line is close to one, it means that there is a high probability that random information is embedded in the LSB plane of the host image. In contrast, if it is close to zero, it means that there is no evidence that the host image hides the information. The green curve in the figure represents the average value of the LSBs on the current block. If the green curve of the stego-image is similar to that of the host image, the embedded information is probably not random. As shown in Fig. 12, the results of Chi-square test obtained by the proposed algorithm are satisfactory. The red line in the figure is flat at zero except for some points at the beginning, and the green curve is very similar to that obtained by the original host image. Hence, we can claim that the proposed algorithm is resistant against Chi-square test.

### *5.3.2 RS attack*

RS attack is a famous stegano-analysis technique proposed by Fridrich et al. [40] that tries to find a steganograpghic message within the LSB plane of a stego-image. This technique can effectively discover the embedding rate of a stego-image. If the embedding rate obtained by the RS attack is less than 0.05 %, the RS attack cannot recognize if data are embedded in the image. To apply the RS attack, we use StegScrete [41] software. Figure 13 shows the results of RS attack obtained by the proposed algorithm and Yu et al.'s method where the first type of hiding is applied, the secret image is "Airplane" and the Host images are "Airplane," "Baboon," "House," "Lena," "Peppers" and "Sailboat," respectively. As shown in Fig. 13, in contrast with Yu et al.'s method that is not immune to RS attack, the proposed algorithm is mostly immune to RS attack. More specifically, the results of RS attack obtained by the proposed algorithm is close to those

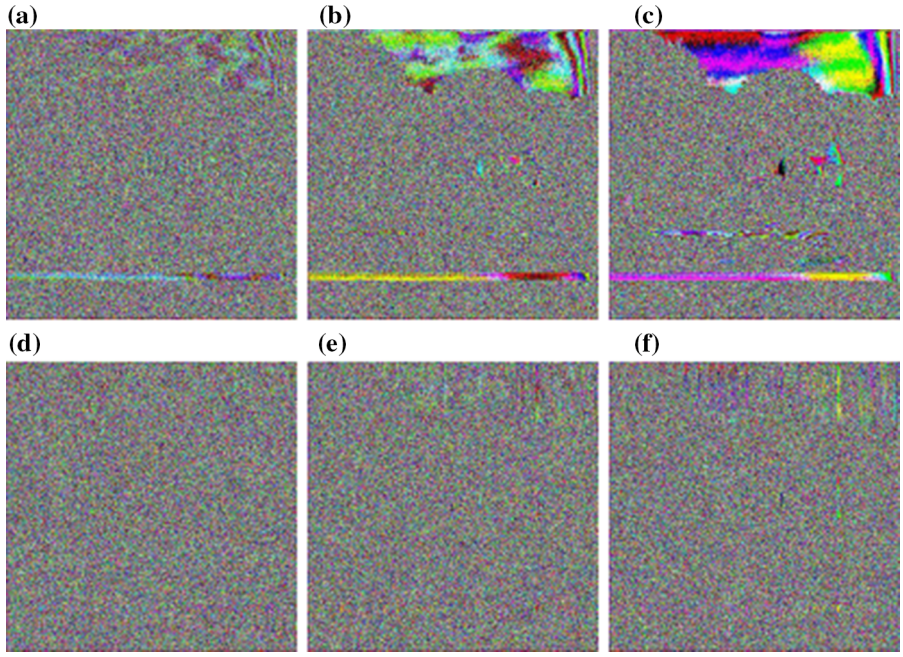


**Fig. 14** The result of byte attack obtained by the proposed algorithm. **a** The result of byte attack to first bit plane of original image "House," **b** the result of byte attack to first bit plane of original image "House," **c** the result of byte attack to first bit plane of original image "House," **d** the result of byte attack to first bit plane of stego-image "House," **e** the result of byte attack to first bit plane of stego-image "House." **f** The result of byte attack to first bit plane of stego-image "House"

obtained by the original host image, and it is only for host image "Baboon" that the RS attack can realize that the image contains a secret image.

### 5.4 Byte attack

The LSB plane of digital images is often random and includes no specific structure. Byte attack technique tries to examine the LSB plane of stego-image to realize if there is a specific pattern in the stego-image. To apply the byte attack, we use StegScrete [41] software. Figure 14 shows three LSB planes of the stego-image "House" where the color secret image "Lena" is embedded. In the figure, if the secret image exists in the host image, we can see the rough picture of the image in the LSB plane.

As seen from Fig. 14, the result of byte attack obtained by the proposed algorithm is promising as there is no specific pattern in the LSB plane of the host image.

## 6 Conclusion

Using cycling chaos system in conjunction with pseudo-random number generators, in this paper, we propose a new robust, efficient and high hiding capacity steganographic algorithm. The proposed algorithm is utilized for embedding both grayscale and color image in a color image. The pseudorandom generators determines in what color pixel position (the row and column) of the host image, the secret data is embedded. The cycling chaos supplies the seeds of RNGs. The proposed algorithm is compared with Yu et al. and Lin et al.'s hiding methods. The results indicate that the proposed algorithm outperforms other algorithms in terms of PSNR values, quality index and hiding capacity. The proposed algorithm was also challenged against visual, RS and byte attacks and Chi-square test. The results show that the proposed algorithm is immune against such attacks.

There are several reasons for why our proposed algorithm has achieved such good performance. This could be attributed to the properties of chaotic signals. The first property is that these signals are unpredictable; therefore, the output of an algorithm that is built based upon these systems is very unpredictable, a property that makes it hard for an intruder to steal information. Second property is that as shown in this paper, a very small change in the parameters results in huge changes in the signal and thus huge changes in the retrieved information. This means that the intruder has to have the exact key to be able to steal information. The third property is that the algorithm has many parameters; thus, the key length could be very large. This property, combined with the second property, means that the key length could be thousands of bits. In terms of the performance, since the three last bits of the pixels are used to store the hidden information, after the data are hidden in the host image, the quality of the host image is very good; thus, an intruder would not suspect that data are hidden in the image. And finally, the process of generating chaotic signals is very fast resulting in a very efficient algorithm.

Note that some other chaotic systems might also be used. For example, Lorenz [42] chaotic systems also show many properties of the chaotic system shown in this paper. They may offer some other tools for an algorithm designer, the study of which remains for future work.

## References


1. Chang, C.-C., Lin, C.-Y., Wang, Y.-Z.: New image steganographic methods using run-length approach. Inf. Sci. **176**(22), 3393–3408 (2006)
2. Liu, C.-L., Liao, S.-R.: High-performance jpeg steganography using complementary embedding strategy. Pattern Recogn. **41**(9), 2945–2955 (2008)
3. Kanso, A., Own, H.S.: Steganographic algorithm based on a chaotic map. Commun. Nonlinear Sci. Numer. Simul. **17**(8), 3287–3302 (2012)
4. Lou, D.-C., Hu, C.-H.: Lsb steganographic method based on reversible histogram transformation function for resisting statistical steganalysis. Inf. Sci. **188**, 346–358 (2012)
5. Yu, Y.-H., Chang, C.-C., Lin, I.-C.: A new steganographic method for color and grayscale image hiding. Comput. Vis. Image Underst. **107**(3), 183–194 (2007)
6. Chang, C.-C., Tai, W.-L., Lin, C.-C.: A reversible data hiding scheme based on side match vector quantization. IEEE Trans. Circuits Syst. Video Technol. **16**(10), 1301–1308 (2006). doi:10.1109/TCSVT.2006.882380
7. Dumitrescu, S., Wu, X.: A new framework of lsb steganalysis of digital media. Signal Process. IEEE Trans. **53**(10), 3936–3947 (2005)
8. Palacios, A., Juarez, H.: Cryptography with cycling chaos. Phys. Lett. A **303**(5–6), 345–351 (2002)
9. Baptista, M.: Cryptography with chaos. Phys. Lett. A **240**(1–2), 50–54 (1998)
10. Sabery, K., Yaghoobi, M.: A simple and robust approach for image hiding using chaotic logistic map. In: Advanced Computer Theory and Engineering, 2008. ICACTE '08. International Conference on, pp. 623–627 (2008)
11. Bassham, III, L.E., Rukhin, A.L., Soto, J., Nechvatal, J.R., Smid, M.E., Barker, E.B., Leigh, S.D., Levenson, M., Van-

gel, M., Banks, D.L., Heckert, N.A., Dray, J.F., Vo, S.: Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, Gaithersburg, MD, USA (2010)
12. Lin, M.H., Hu, Y.C., Chang, C.C.: Both color and gray scale secret images hiding in a color image. Int. J. Pattern Recogn. Artif. **16**(6), 697–713 (2002)
13. Sharma, G.: Digital Color Imaging Handbook. CRC Press, Boca Raton (2002)
14. Orchard, M., Bouman, C.: Color quantization of images. IEEE Trans. Signal Process. **39**(12), 2677–2690 (1991)
15. Heckbert, P.: Color image quantization for frame buffer display. SIGGRAPH Comput. Graph. **16**(3), 297–307 (1982)
16. Wan, S.J., Prusinkiewicz, P., Wong, S.K.M.: Variance-based color image quantization for frame buffer display. Color Res. Appl. **15**(1), 52–58 (1990)
17. Lo, K.C., Chan, Y.H., Yu, M.P.: Colour quantization by three-dimensional frequency diffusion. Pattern Recogn. Lett. **24**(14), 2325–2334 (2003)
18. Sirisathitkul, Y., Auwatanamongkol, S., Uyyanonvara, B.: Color image quantization using distances between adjacent colors along the color axis with highest color variance. Pattern Recogn. Lett. **25**(9), 1025–1043 (2004)
19. Celebi, M., Wen, Q., Hwang, S.: An effective real-time color quantization method based on divisive hierarchical clustering. J. Real-Time Image Process. 1–16 (2012)
20. Su, Q., Hu, Z.: Color image quantization algorithm based on self-adaptive differential evolution. Intell. Neurosci. 2013, 3:3–3:3 (2013)
21. Goldberg, N.: Colour image quantization for high resolution graphics display. Image Vis. Comput. **9**(5), 303–312 (1991)
22. Thomas, S.W.: Efficient inverse color map computation. In: Arvo, J. (ed.) Graphics Gems II, pp. 116–125. Academic Press, Waltham (1991)
23. Cachin, C.: An information-theoreticmodel for steganography. Inf. Comput. **192**(1), 41–56 (2004)
24. Tadiparthi, G.R., Sueyoshi, T.: A novel steganographic algorithm using animations as cover. Decis. Support Syst. **45**(number), 937–948 (2008)
25. Moulin, P., O'Sullivan, J.: Information-theoretic analysis of information hiding. IEEE Trans. Inf. Theory **49**(3), 563–593 (2003)
26. Malini, M., Anurenjan, P.: A new algorithm for data hiding in images using contourlet transform (2011)
27. Liao, S.-R., Liu, C.-L.: High-performance jpeg steganography using complementary embedding strategy. Pattern Recogn. **45**, 2945–2955 (2008)
28. Zhao, Y., Zhao, N., Ren, G., Zhang, B.: A novel large capacity image hiding method based on the orthogonal chaotic sequences. In: Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP '06. International Conference on, pp. 613–616 (2006)
29. Tsai, D.-S., Horng, G., Chen, T.-H., Huang, Y.-T.: A novel secret image sharing scheme for true-color images with size constraint. Inf. Sci. **179**(19), 3247–3254 (2009)
30. Multiprecision computing toolbox. http://www.advanpix.com/
31. Mazloom, S., Eftekhari-Moghadam, A.M.: Color image encryption based on coupled nonlinear chaotic map. Chaos Solitons Fractals **42**(3), 1745–1754 (2009)
32. Wang, Y., Wong, K.-W., Liao, X., Chen, G.: A new chaos-based fast image encryption algorithm. Appl. Soft Comput. **11**(1), 514–522 (2011)
33. University of Southern California, Standard test images, volume 3: miscellaneous. http://sipi.usc.edu/database/database.php?volume=misc
34. Rongrong, N., Qiuqi, R.: Embedding information into color images using wavelet. In: TENCON '02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, vol. 1, pp. 598–601 (2002)
35. Liu, T., Zheng-ding, Q.: A dwt-based color image steganography scheme. In: Signal Processing, 2002 6th International Conference on, vol. 2, pp. 1568–1571 (2002)
36. Wang, Z., Bovik, A.: A universal image quality index. IEEE Signal Process. Lett. **9**, 81–84 (2002)
37. Ghebleh, M., Kanso, A.: A robust chaotic algorithm for digital image steganography. Commun. Nonlinear Sci. Numer. Simul. **19**(6), 1898–1907 (2014)
38. Provos, N., Honeyman, P.: Detecting steganographic content on the internet. Technical report. In: ISOC NDSS02 (2001)
39. El-Loco, G.: A few tools to discover hidden data. http://www.guillermito2.net/stegano/tools/index.html
40. Fridrich, J., Goljan, M., Du, R.: Detecting LSB steganography in color and gray-scale images. IEEE MultiMedia **8**(4), 22–28 (2001)
41. Muoz, A.: A simple steganalysis tool: StegSecret. http://stegsecret.sourceforge.net
42. Lian, S.: Multimedia Content Encryption: Techniques and Applications, 1st edn. Auerbach Publications, Boston (2008)