Optimal Image Based Information Hiding with One-dimensional Chaotic Systems and Dynamic Programming

Yinglei Song[1], Jia Song[2], and Junfeng Qu[3]
[1]School of Electronics and Information Science, Jiangsu University of Science and Technology, China
[2]Department of Electronic and Information Technology, Suzhou Vocational University, China
[3]Department of Computer Science and Information Technology, Clayton State University, USA

Abstract: *Information hiding is a technology aimed at the secure hiding of important information into digital documents or media. In this paper, a new approach is proposed for the secure hiding of information into gray scale images. The hiding is performed in two stages. In the first stage, the binary bits in the sequence of information are shuffled and encoded with a set of integer keys and a system of one-dimensional logistic mappings. In the second stage, the resulting sequence is embedded into the gray values of selected pixels in the given image. A dynamic programming method is utilized to select the pixels that minimize the difference between a cover image and the corresponding stego image. Experiments show that this approach outperforms other information hiding methods by 13.1% in Peak Signal to Noise Ratio (PSNR) on average and reduces the difference between a stego image and its cover image to 0 in some cases.*

Keywords: *Encryption and hiding, minimized hiding effects, improved security, convenient recovery.*

Received June 3, 2019; accepted February 15, 2021
https://doi.org/10.34028/iajit/19/1/1

1. Introduction

In the past two decades, a tremendous amount of digital data and information was generated along with the significant development and achievements in information technology [9]. The protection of the intellectual property rights associated with some important digital data and information has thus become an important issue in information science [3, 24]. Numerous methods and techniques have been developed to resolve this issue, including information hiding technology [9]. Specifically, information hiding technology hides important data and information within digital documents such as images, videos and files, the resulting documents are required to have high similarity with the original ones [1]. A document where the hiding is performed is a cover media and the one that contains the hidden information is a stego media [6]. In general, human eyes are unable to recognize the difference between a cover media and the corresponding stego media [26].

Since the transmission and storage of multimedia data are often based on images, embedding information into gray scale images has become an important approach for information hiding [1, 12, 14]. Researchers have developed a large number of methods that can hide information into gray scale images [6, 12, 13, 14, 26, 27], such as the side match technology [5], the hiding technology based on pixel difference expansion and modulus function [26], and the information hiding technology based on pixel value difference and Least Significant Bit (LSB) replacement [27]. Most of these methods select a number of neighboring pixels in the image based on the difference in gray values. The gray values of these neighboring pixels are changed to contain the data that needs to be embedded.

Recently, chaotic systems have been used in a large number of algorithms to encrypt images [2, 3, 5, 8, 10, 11, 29]. A question thus immediately arises on whether chaotic systems can be applied to improve the security of an information hiding algorithm.

The security of an approach can be further enhanced if the information is encrypted with a robust encryption method before it is hidden within an image [18]. An encryption method based on chaotic systems can possibly provide improved security for an image based information hiding approach [3, 7]. In addition, it is clear that the difference between a cover image and its corresponding stego image is an important measure of the security of an approach for information hiding [6]. So far, most of the existing approaches have not considered the minimization of this difference for optimal effects of information hiding [19, 21, 22].

In this paper, a new image based information hiding approach is proposed to hide a sequence of binary bits into a gray scale image with improved security. A number of integer keys and a system of one-dimensional logistic mappings are used to encrypt the sequence before it is hidden into an image. In addition, the hiding of information is performed with a minimized difference between a cover image and the corresponding stego image.

The approach performs hiding in two stages. In the first stage, it divides the sequence into subsequences of equal length. The location of a bit in the sequence is represented by a pair of integers, one of them corresponds to the subsequence where the bit is located and the other one is the relative position of the bit in the subsequence. All bits in the sequence are thus mapped to a grid in two-dimensional space and the bits are shuffled and encoded based on a number of integer keys and a system of one-dimensional logistic mappings. Specifically, each row and column in the grid is associated with a positive integer key. The relocation of bits is performed row by row first. A bit in a row of the two-dimensional grid is relocated based on the product of the integer key of the row and the current position of the bit in the row. After a row-based relocation is complete, bits are grouped in columns and the bits in each column are relocated based on the integer key of the column.

In the second stage of the approach, the shuffled and encoded sequence is divided into regions of equal length and the hiding of the bits is completed by sequentially embedding the regions into the gray scale image. To obtain a stego image that has the highest similarity to the cover image, a measure that evaluates the difference between a cover image and its stego image is developed and the locations for embedding can be determined by a dynamic programming method that can minimize the measure. The embedding of a region is performed by substituting the least significant bits of the gray value of the pixel in the determined location with the bits in the region.

The keys for recovering the embedded sequence include the integer keys associated with rows and columns of the grid for shuffling, the one-dimensional logistic mappings for encoding and the integers needed to determine the locations of embedding. The proposed approach is able to recover the hidden information without the cover image or a standard image. A simple analysis shows that the approach has a key space of large size and the hidden information is thus secure against attacks based on exhaustive search.

The contributions of the proposed approach can thus be summarized from two different aspects. Firstly, an encryption method based on chaotic systems is proposed to encrypt the information before it is hidden into an image. This step of encryption can protect the content of the information from being identified even in cases where the information hidden in a stego image has been retrieved by the adversary side. Secondly, the difference between a cover image and its corresponding stego image is minimized when the encrypted information is hidden within the image. This minimization step can further enhance the security of the information hidden within a stego image.

Experiments show that this approach can generate excellent information hiding results on medical images. In addition, a comparison with two other existing methods on a large number of images for image based information hiding shows that the proposed approach can generate stego images with significantly higher similarity values to the corresponding cover images.

2. Related Work

Numerous methods have been developed to improve the performance of information hiding. For example, a multi-level hiding strategy is developed in [17] to achieve larger hiding capacity while maintaining the high similarity between a stego image and its cover image. In [1], an information hiding algorithm is developed based on histogram shifting in efficient compressed domain. In [22], a deep learning based technique is used to choose the appropriate carrier for real-time image data hiding. In [21], an adaptive threshold generation mechanism is proposed to determine the thresholds needed for the pixel value ordering hiding scheme. In [19], a new turtle-shell based information hiding algorithm is developed to improve the capacity of embedding while maintaining good image quality. In [28], a reversible data hiding algorithm based on block truncation coding is developed. In [13], the degradation of medical images after a frequency domain based hiding algorithm hides data within them is studied. In [12], two novel variations of the classical histogram shift methods are developed to further improve the capacity of embedding in medical images. In [16], a reversible information hiding algorithm that uses adaptive block truncation coding along with an edge based quantization method is proposed. In [18], a reversible data-hiding approach is proposed based on redundant space transfer. The proposed approach transfers redundant space from an original image to its encrypted image. A general reversible data hiding algorithm is then applied for information hiding. In [20], a new information hiding algorithm is developed to combine the Chinese Remainder Theorem and a new extraction function to enlarge the capacity of embedding.

Most of the existing methods can successfully hide information into a gray scale image while maintaining a high similarity between a cover image and the corresponding stego image [6, 12, 13, 14, 26, 27]. However, most of these methods are unable to fully utilize the pixels in an image to enhance the similarity between a stego image and its cover image [12], which may not be desirable for certain applications [13, 14]. In addition, the information embedded into an image can often be directly obtained by processing a stego image or comparing a stego image with its cover image [27], the embedded information is thus not secure when the stego image or both the cover image and its stego image need to be transmitted through the internet [6].

Recently, many encryption methods based on chaotic systems have been proposed. In [15], an algorithm that can encrypt color images by cellular automata is developed. The encryption of a colored image is performed with a hybrid hyper-chaotic system. In [25], an algorithm is developed to encrypt an image by combing a bit level permutation architecture with chaotic systems. In [29], the avalanche effect of a high-dimensional chaotic system is used for the encryption of images. The work in [23] uses two sets of one-dimensional logistic systems to generate a robust encryption of an image.

3. The Proposed Method

The information that needs to be hidden within a gray scale image can be represented by a sequence of binary bits. The hiding of such a sequence into a gray scale image is performed in two stages. In the first stage, the binary bits in the sequence are first relocated and then encoded with a system of one-dimensional logistic systems. In the second stage, the encrypted sequence is hidden into a cover image with a dynamic programming method. The difference between a cover image and its stego image is minimized to achieve the optimal effects of hiding.

The steps in the first stage of the proposed approach can thus be sketched as follows.

1. Change the locations of the bits in the sequence with a set of positive integer keys.
2. The sequence obtained in step 1 is encoded with a number of one-dimensional logistic systems.
3. Return the sequence obtained in step 2 as the shuffled and encoded sequence.

The second stage of the proposed approach contains the following steps.

1. Divide the shuffled and encoded sequence into a number of regions of equal length;
2. Use a dynamic programming method to select pixels for hiding.
3. Substitute the least significant bits in the gray values of selected pixels with the bits in the regions of the shuffled and encoded sequence.
4. Return the resulting image as the stego image that contains the hidden information.

The parameters utilized in both stages are needed for the recovery of the hidden information from a stego image. The steps of the recovery process can be described as follows.

1. Find the selected pixels in the stego image with the parameters used in the second stage.
2. Combine the least significant bits of the gray values of the selected pixels into the shuffled and encoded sequence.
3. Use the parameters of the chaotic systems to decode the sequence obtained in step 2.
4. Use the integer keys for shuffling to relocate the binary bits in the sequence obtained in step 3 to recover the original sequence.
5. Return the sequence obtained in step 4 as the recovered information.

3.1. The Shuffling of the Sequence

Let S be a sequence of binary bits that need to be hidden into a given gray scale image. L_S denotes the length of S and $S(i)$ is the i th bit in S. Based on a positive integer c such that $c<L_S$ and $r=\lfloor L_S/c \rfloor+1$, S can be sequentially divided into r subsequences such that at least r-1 of these subsequences are of equal length and each of them contains c bits. The number of bits in the last subsequence could be less than c. If it is the case, a number of bits of 0 can be padded to the end of S such that the length of the last subsequence is also c. It is thus assumed that S is sequentially divided into r subsequences of length c in the rest of the paper.

Based on the subsequences in S, the position of $S(i)$ can be described by a pair of integers (u,v) where $u=\lfloor i/c \rfloor$ and $v=i$ mod c. u is the row number of $S(i)$ and v is its column number. Each bit in S can thus be uniquely mapped to a point in a two-dimensional space. All bits with the same row number form a row. Similarly, all bits with the same column number form a column. It is thus clear that all bits with the same row number are in the same row, and those with the same column number are in the same column. The rows formed by bits in S are numbered by integers 1,2…, r and columns can be numbered by 1,2…, c.

Each row m, where $1\leq m \leq r$, is then associated with a positive integer k_m such that the greatest common divisor of c and k_m is 1. Similarly, each column n, where $1\leq n\leq c$, is associated with a positive integer l_n such that the greatest common divisor of l_n and r is 1. As the first step of shuffling, each bit is relocated within its row. Specifically, the bit mapped to an integer pair (s,t) is relocated to $(s,g(t,k_s))$, where $g(t,k_s)$ is computed as follows.

$$g(t,k_s)=(k\times t)\bmod c \tag{1}$$

It is straightforward to see that for two arbitrary different integers t_1 and t_2 such that $0\leq t_1< c$ and $0\leq t_2 <c$, $g(t_1.k_s)$ and $g(t_2,k_s)$ are also different. Indeed, if it is not the case, there exist two different integers t_1 and t_2 such that $0\leq t_1 <c$ and $0\leq t_2 <c$, and $g(t_1,k_s)= g(t_2,k_s)$. From Equation (1), the following equation must hold for t_1 and t_2.

$$k_s(t_1-t_2)=cp \tag{2}$$

Where p is an integer. However, since the greatest common divisor of c and k_s is 1, Equation (2) implies that t_1- t_2 must be divisible by c, which is contradictory to the fact that $|t_1$-$t_2| <c$. $g(t_1.k_s)$ and $g(t_2,k_s)$ thus must be different. This fact ensures that no two bits are relocated to the same position and no conflicts would occur in the first step of the designed relocation procedure.

In the second step of the shuffling, relocations are performed in columns. Each bit is relocated within its column. Specifically, the bit that is currently in row x and column y is relocated to $(h(x,l_y),y)$, where $h(x,l_y)$ is computed as follows.

$$h(x,l_y)=(x\times l_y) \bmod r \tag{3}$$

Similarly, based on the same argument that has been shown above, no two bits are relocated to the same position and no conflicts would occur in the second step of the shuffling process. In the final step of the shuffling process, each bit is mapped from its position in the two dimensional space back to the corresponding position in the sequence. Specifically, the bit currently in position (u,v) in the two dimensional space is mapped back to position $uc+v$ in the sequence.

3.2. The Encoding of the Shuffled Sequence

Let R be the shuffled sequence generated from the original sequence S by the shuffling process described in subsection 2.1. The length of R is also L_S. Based on a given positive integer parameter b, where $1\leq b\leq 8$, R can be sequentially divided into regions of length b. Here, L_S is assumed to be divisible by b. Since the bits in each region form the binary encoding of an integer between 0 and 2^b-1, a sequence of integers $w_1,w_2,\ldots,w_a$ are used to represent R, where $a=L_S/b$.

Based on a one dimensional logistic mapping G_μ, a sequence of real numbers z_0, $z_1,\ldots$, z_k, $z_{k+1},\ldots$, where $k\geq 0$, can be generated based on the following recursion.

$$z_{k+1}=\mu z_k(1-z_k) \tag{4}$$

Where μ is a parameter of the recursion and must satisfy $0< \mu <4$. It is clear that $0< z_k <1$ holds for each $k\geq 0$ if the initial value z_0 of the sequence is a real number between 0 and 1. In other words, all numbers in the sequence are positive and less than 1 when the initial value of the sequence is a real number between 0 and 1.

It is well known that a logistic mapping has the property of a one-dimensional chaotic system. Specifically, when $3.57 \leq \mu <4$, the period of the numbers in the sequence becomes infinitely large and the numbers in the sequence are sensitive to the initial value z_0 [7]. When k is large enough, a significantly different z_k would arise from a perturbation of a tiny amount in z_0. The initial value z_0 can thus be used as a key to generate encoded data from a one-dimensional logistic mapping. In fact, many existing image encryption algorithms utilize the chaotic property of one-dimensional logistic mappings to encrypt images.

To perform the encoding of R, a system of d one-dimensional logistic mappings $G_{\mu_1}, G_{\mu_2},\ldots, G_{\mu_d}$ is used with different parameters μ_1, $\mu_2,\ldots,\mu_d$. Each logistic mapping is assigned a different initial value and a sequence of real numbers can be generated for each logistic mapping. Let $z_{1,0}$, $z_{2,0},\ldots$, $z_{d,0}$ be the initial values for $G_{\mu_1}, G_{\mu_2},\ldots, G_{\mu_d}$ respectively.

A large enough integer k_0 is then selected such that the chaotic behavior of a one-dimensional logistic mapping starts to be significant when more than k_0 numbers have been generated in the sequence. To encode w_i $(1\leq i\leq a)$ in R, d integers $\lambda_{i,1},\lambda_{i,2},\ldots,\lambda_{i,d}$ are generated as follows.

$$\lambda_{i,j}=(z_{j,i+k_0}\times M) \bmod 2^b \tag{5}$$

Where $1\leq j\leq d$, M is a large integer and $z_{j,i+k0}$ is the $i+k_0$ th number generated with G_{μ_j} and an initial value of $z_{j,0}$.

From $\lambda_{i,1},\lambda_{i,2},\ldots,\lambda_{i,d}$, w_i can be encoded based on the following recursive relation.

$$\kappa_{i,q+1}=\kappa_{i,q}\oplus\lambda_{i,q+1} \tag{6}$$

Where $0\leq q \leq d$-1, $K_{i,0} = w_i$ and $K_{i,d}$ is the encoded result for w_i.

After all of the integers w_1, $w_2,\ldots,w_a$ have been encoded, the binary forms of $K_{1,d}$, $K_{2,d}\ldots,K_{i,d}$ are sequentially combined into a sequence of binary bits. The resulting sequence is the shuffled and encoded sequence of S.

3.3. The Embedding of the Shuffled and Encoded Sequence

Let B_2 denote the shuffled and encoded sequence obtained with the approaches in Subsections 2.1 and 2.2. Let I be the gray scale image where B_2 needs to be embedded. There are m rows and n columns in I and $I(g_p,h_p)$ is the gray value for the pixel in row g_p and column h_p in I. In the second stage, the bits in B_2 are sequentially embedded into I.

The bits in B_2 are embedded into I in groups and the numbers of bits contained in all groups are equal. Let w_d be the number of bits in each group. Since the gray value of a pixel contains 8 bits, the inequality $1\leq w_d \leq 8$ must hold for w_d. The bits in B_2 are thus sequentially divided into $\beta=\lfloor cr/w_d\rfloor+1$ subsequences, the bits in each subsequence form a group and each subsequence contains w_d bits. Integers $1,2,\ldots,\beta$ are used to sequentially number the subsequences. Let e and f be positive integers that satisfy $e<\lfloor mn/5\beta\rfloor$ and $f\leq mn-be$, the bits in subsequence $l(1\leq l\leq\beta)$ is embedded into the gray value of pixel (g_l,h_l)

in I, where g_l and h_l are computed with Equations (7) and (8). It is shown later that the values of e and f can be determined by a dynamic programming method.

$$g_l = \lfloor (f + (l-1)e)/n \rfloor + 1 \qquad (7)$$

$$h_l = (f + (l-1)e) \bmod n \qquad (8)$$

The gray value of pixel (g_l,h_l) is updated to include the w bits in subsequence l as follows.

$$I_2(g_l, h_l) = \lfloor I(g_l, h_l)/2^{w_d} \rfloor + C_l \qquad (9)$$

Where $I_2(g_l,h_l)$ is the gray value of pixel (gl,hl) after theembedding is performed and C_l is the value representedby the wd bits in subsequence l. It is clear from Equation (9) that the method replaces wd least significant bits in the gray value of pixel (g_l,h_l) by the bits in subsequence l to complete the embedding.

The values of e and f can be determined as follows. In principle, e and f should be selected to minimize the difference between the stego image and the cover image. The difference between the cover image I and the stego image I_2 can be evaluated by $D(I,I_2)$ as defined in Equation (10). In practice, a reasonable value for e can be selected and $D(I,I_2)$ is minimized to obtain a value for f.

$$D(I, I_2) = \sum_{i=1}^{m} \sum_{j=1}^{n} | I(i, j) - I_2(i, j) | \qquad (10)$$

Specifically, given a fixed e, the value of f can be determined by minimizing $D(I,I_2)$. The minimization can be performed with a dynamic programming approach as follows.

A two dimensional table $T(o,s)$ is maintained to store the difference between I and I_2 in the case where the value of f is equal to o and subsequences from 1 to s have been embedded into I. It is straightforward to see that $T(o,s+1)$ can be computed based on the recursive relation shown in Equations (11) and (12).

$$D(s+1) = | I(g_{s+1}, h_{s+1}) - I_2(g_{s+1}, h_{s+1}) | \qquad (11)$$

$$T(o, s+1) = T(o, s) + D(s+1) \qquad (12)$$

Where g_{s+1} and h_{s+1} can be computed as follows.

$$g_{s+1} = \lfloor (f + so)/n \rfloor + 1 \qquad (13)$$

$$h_{s+1} = (f + so) \bmod n \qquad (14)$$

The value of $T(o,s)$ for any o that satisfies $1 < o \le mn-be$ is set to be zero. The value of f that can minimize $D(I,I_2)$ is determined by Equation (15). Specifically, the integer o that minimizes $T(o,s)$ is assigned to f.

$$f = \underset{1 \le o \le mn-be}{\arg\min} \{T(o,b)\} \qquad (15)$$

3.4. The Recovery of the Embedded Sequence

Let I_2 be the resulting stego image after the embedding of B_2 into I is complete. Given the values of c, r, w_d, e, f, the initial values and parameters of the different one-dimensional logistic mappings for encoding, and the integer keys used in the shuffling of the original sequence S, S can be recovered from I_2 in two stages. In the first stage, a set of pixels whose gray values contain the bits in sequence B_2 can be efficiently determined. The w_d least significant bits of the gray values of these pixels can then be extracted and combined sequentially to reconstruct. B_2 Specifically, for each integer l where $1 \le l \le \beta$, the binary value encoded by the w_d bits in subsequence l in B_2 can be computed with Equation (16)

$$C_l = I_2(g_l, h_l) \bmod 2^{w_d} \qquad (16)$$

Where g_l and h_l are computed with Equations (7) and (8). C_l is the value represented by the w_d bits in subsequence l in B_2. Subsequence l in B_2 can then be determined from C_l.

In the second stage, the original sequence S can be reconstructed from B_2. Let μ_1, μ_2,…,μ_d. be the initial values of the logistic mappings $G_{\mu_1}, G_{\mu_2}, ..., G_{\mu_d}$ used for the encoding of the shuffled sequence R, and $z_{1,0}$, $z_{2,0}$,..., $z_{d,0}$ be their initial values. Since R can be represented by a sequence of integers $w_1, w_2, ..., w_a$, where each integer is represented by b binary bits and the encoding of R is performed by sequentially encoding the integers in the sequence, the recovery of R from B_2 can also be performed sequentially such that each individual integer in R can be recovered first and the recovered integers are then combined to reconstruct R. B_2 is thus sequentially divided into a regions of length b and $\delta_1, \delta_2, ..., \delta_a$ are used to denote the resulting sequence of integers.

To recover w_i ($1 \le I \le a$), d integers $\lambda_{i,1}$,$\lambda_{i,2}$,…,$\lambda_{i,d}$, are first computed from $G_{\mu_1}, G_{\mu_2}, ..., G_{\mu_d}$ by Equation (5). Based on $\lambda_{i,1}$,$\lambda_{i,2}$,…,$\lambda_{i,d}$, w_i can be recovered from δ_i by the following recursive relation.

$$\phi_{i,q+1} = \phi_{i,q} \oplus \lambda_{i,d-q} \qquad (17)$$

Where $0 \le q \le d$-1, $\phi_{i,0} = \delta_i$ and $\phi_{i,d}$ is the result of decoding. In other words, $w_i = \phi_{i,d}$.

After all of the integers $w_1, w_2, ..., w_a$ have been recovered, R can be reconstructed by sequentially combining the binary forms of w_1,w_2,…,w_a into a single sequence.

The last step of recovery is to relocate the bits in R to recover the original sequence S. The recovery of S is based on the integer keys associated with all columns and rows in the two-dimensional grid where the bits are mapped to during the shuffling process. To describe this clearly, k_1,k_2,…,k_r are used to denote the integer keys associated with the rows and l_1,l_2,…,l_c denote the integer keys for the columns.

Bits in R are first mapped to a two-dimensional grid with *r* rows and *c* columns. Specifically, the *i* th bit in *R* is mapped to the point with a row number of $\lfloor i/c \rfloor$and a column number of *i* mod *c*. The bits are then relocated by columns after the mapping is completed. Each bit is relocated within its column. Specifically, the bit currently in row $h(x,l_y)$ and column *y* is relocated to (x,y), where $h(x,l_y)$ is computed with Equation (3). To complete the relocation for all bits in column *y*, for each *x* that satisfies $1\leq x \leq r$, the value of $h(x,l_y)$ is calculated and the bit currently in location $(h(x,l_y),y)$ is relocated to(x,y).

The bits are then relocated by rows. Each bit is relocated within its row. Specifically, the bit currently in $(s,g(t,k_s))$ is relocated to (s,t), where $g(t,k_s)$ is computed with Equation (1). To complete the relocation for all bits in row s , for each t that satisfies $1\leq t \leq c$, the value of $g(t,k_s)$ is computed and the bit currently in $(s,g(t,k_s))$ is relocated to (s,t). The resulting sequence is the original sequence S . It is clear that the recovery of the embedded information can be performed without the cover image or a standard image.

4. Experimental Results and Discussions

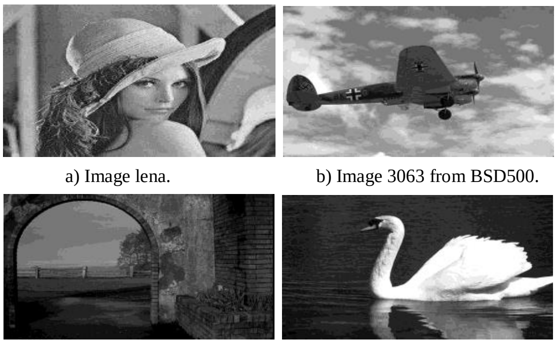
The proposed approach has been implemented into a computer program in MATrixLABoratory (MATLAB). Its performance is tested by hiding information into gray scale images obtained from benchmark datasets. The experiments consist of two parts. In the first part, the overall performance of the proposed approach is evaluated with a number of ordinary and medical images. In the second part, the proposed approach is tested on all images in a benchmark dataset and its performance is compared with that of two other existing methods.

In all experiments, the smallest *c* positive integers coprime with *c* are used to relocate the bits for each column. Similarly, the smallest positive integers coprime with *r* are applied for the relocation of bits in each row. The value of *c* is chosen to be 20 for all experiments.

For the encoding of a shuffled sequence, five chaotic systems are used in all experiments. The parameters for the chaotic systems are real numbers between 3.7 and 3.9. These parameters are obtained by a linear interpolation with the number of chaotic systems used for encoding. The initial values of the one-dimensional logistic mappings are numbers randomly generated between 0 and 1 and *M* is set to be 10^6. The value of w_d is varied during the experiments for a comprehensive evaluation of the proposed approach.

4.1. On Benchmark and Medical Images

The performance of the proposed approach is first tested by hiding information into four benchmark ordinary gray scale images and two medical images. One of the ordinary benchmark images is the well known benchmark image Lena. The other three of the ordinary benchmark images are selected from the Berkeley Segmentation Data Set and Benchmarks 500 (BSDS500) at [4] (downloaded from the website at Jan 21, 2018). The two medical images are both obtained from the database of The Cancer Imaging Archive (TCIA) public access at [8] (downloaded from the website at June 11, 2018).



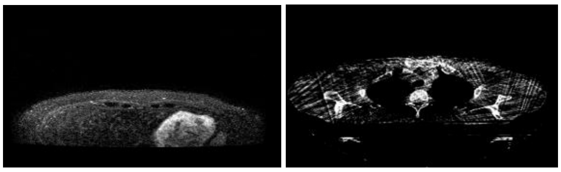
a) Image lena.

b) Image 3063 from BSD500.

c) Image 5096 from BSD500.

d) Image 8068 from BSD500.

Figure 1. Ordinary benchmark images for testing the proposed approach.



a) Med1.

b) Med2.

Figure 2. Medical images for testing the proposed approach.

*Both medical images are downloaded from the cancer imaging archive (TCIA) public access.

The four testing ordinary benchmark images are shown in Figure 1-a), 1-b), 1-c), and 1-d) and the two testing medical images are shown in Figure 2-a) and 2-b). All testing images are scaled to the same size, which is 100×100.

Two measures are used to evaluate the difference between a cover image and the corresponding stego image. One of them is the Square Sum of the Difference (SSD) in the gray values of the corresponding pixels in both images. Given a cover image I_c and its corresponding stego image I_s, both images contain *m* rows and *n* columns. The SSD between I_c and I_s can be computed as follows.

$$SSD(I_c, I_s) = \sum_{i=1}^{m}\sum_{j=1}^{n} | I_c(i,j) - I_s(i,j) |^2 \qquad (18)$$

It is clear that a higher value of SSD suggests a larger amount of difference between the cover image and the corresponding stego one. Another measure is the Peak Signal of Noise Ratio (PSNR) defined between the cover image and the corresponding stego image. In the case where both I_c and I_s contain *m* rows and *n* columns, the PSNR between I_c and I_s can be computed as follows.

$$PSNR(I_c, I_s) = 20\log_{10}(\frac{255\sqrt{mn}}{\sqrt{SSD(I_c, I_s)}}) \quad (19)$$

It can be seen from Equation (2) that a higher value of *PSNR*(I_c,I_s) suggests a higher similarity between I_c and I_s. In the case where I_c and I_s are identical, *SSD*(I_c,I_s) is used as the only measure to represent the difference of I_c and I_s.

Table 1. The means and standard deviations (STD) of SSDs and PSNRs obtained with the proposed approach on the testing images when w_d is 1.

Testing Image	SSD		PSNR	
	Mean	STD	Mean	STD
Lena	46.60	4.64	71.47	0.42
3063	51.20	6.38	71.07	0.57
5096	51.20	3.68	71.05	0.31
8068	52.50	4.04	70.94	0.30
Med1	0.00	0.00	—	—
Med2	0.00	0.00	—	—

Table 2. The means and standard deviations (STD) of SSDs and PSNRs obtained with the proposed approach on the testing images when w_d is 5.

Testing Image	SSD		PSNR	
	Mean	STD	Mean	STD
Lena	563.50	109.76	58.12	2.85
3063	478.60	110.05	58.20	2.23
5096	526.30	127.58	58.54	2.15
8068	300.60	101.86	56.17	1.54
Med1	0.00	0.00	—	—
Med2	0.00	0.00	—	—

The performance of this approach is first evaluated when different values of w_d are used for embedding. The tested values of w_d include 1 and 5. For each given value of w_d, 100 sequences of binary bits are randomly generated and hidden into each testing image. Each sequence contains 300 binary bits. Tables 1 and 2 show the means and standard deviations of the SSDs and PSNRs between each testing image and its stego image when different values of w_d are used for embedding.

It can be seen from Tables 1 and 2 that the similarity between a cover image and the corresponding stego image becomes lower when the word width w_d for embedding increases, which is not out of expectation. In addition, the performance becomes less stable when w_d increases. From Table 2, it is clear that the PSNR values deteriorate significantly when w_d increases from 1 to 5, which suggests that a word width less than 5 should be selected for the proposed approach to achieve satisfactory results in practice. However, an unexpected result is that the hiding of all sequences within the two medical images can be performed without generating any change in cover images, which may suggest that the proposed approach should be used with medical images to achieve the most satisfactory results of hiding in practice.

4.2. A Comparison with Other Methods

The overall performance of this approach is compared with that of two other existing methods, including the methods developed in [6, 26]. Specifically, randomly generated binary sequences are hidden into all ordinary benchmark images and both medical images with the proposed algorithm, the methods developed in [6, 26], and the performance of all three methods is evaluated based on the resulting stego images.

Table 3. The means and standard deviations (STD) of SSDs and PSNRs obtained with three methods on the testing images when the length of the sequence is 200.

Method		8068	Lena	5096	3063	Med1	Med2
The proposed method	PSNR	**73.83**	**73.42**	**73.45**	**73.30**	—	—
	STD	0.45	0.62	0.42	0.36	—	—
	SSD	**27.00**	**29.80**	**29.50**	**30.50**	**0.00**	**0.00**
	STD	2.75	4.32	3.38	3.12	0.00	0.00
Method in [5]	PSNR	65.00	67.38	61.70	70.44	—	—
	STD	0.98	2.50	1.53	0.45	—	—
	SSD	210.10	136.10	467.40	59.10	0.00	1.00
	STD	41.08	70.49	130.32	6.25	0.00	0.94
Method in [24]	PSNR	62.99	63.31	60.96	62.60	57.50	57.44
	STD	0.81	0.89	1.16	1.12	0.63	0.47
	SSD	331.40	309.10	539.00	368.00	1167.60	1179.90

Two different values are selected for sequence length, including 200 and 600. For each given sequence length, 100 sequences of binary bits are randomly generated. The value of w_d is chosen to be 1 for the proposed approach. Tables 3 and 4 compare the means and standard deviations of the SSDs and PSNRs of the stego images obtained with the proposed approach and the other two methods on each given sequence length.

It is clear from Tables 3 and 4 that the proposed approach can achieve perfect hiding on both Med1 and Med2 for all values of sequence length. The mean values of PSNRs achieved by the proposed approach are significantly higher than the other two methods on the four ordinary benchmark images for all sequence lengths. This suggests that the proposed approach consistently outperforms the other two methods on all testing images. In addition, the standard deviations of SSDs and PSNRs confirm that the performance of the proposed approach is reliable and robust. Tables 3, 4, and 5 also show that the method developed in [6] can achieve a perfect embedding of all generated sequences on Med1 and a near perfect embedding of all generated sequences on Med2.

Table 4. The means and standard deviations (STD) of SSDs and PSNRs obtained with three methods on the testing images when the length of the sequence is 600.

Method		8068	Lena	5096	3063	Med1	Med2
The proposed method	PSNR	**67.69**	**67.45**	**67.54**	**67.40**	—	—
	STD	0.18	0.29	0.16	0.34	—	—
	SSD	**110.80**	**117.10**	**114.70**	**118.60**	**0.00**	**0.00**
	STD	4.71	7.77	4.10	9.59	0.00	0.00
Method in [5]	PSNR	60.98	61.45	56.70	66.20	—	—
	STD	0.38	1.06	1.18	0.22	—	—
	SSD	520.70	477.40	1434.80	156.00	0.00	2.80
	STD	45.85	104.58	143.73	8.29	0.00	1.25
Method in [24]	PSNR	58.19	57.94	56.71	57.83	52.59	52.74
	STD	0.57	0.69	0.55	0.55	0.29	0.30
	SSD	994.50	1057.5	1397.9	1079.5	3592.6	3470.0
	STD	104.76	143.13	142.07	124.11	164.36	147.84

Table 5. The means and standard deviations (STD) of SSDs and PSNRs obtained with three methods on the 200 images in the train folder of the BSDS500 dataset. SL is sequence length.

Method		SL=200	SL=300	SL=400	SL=500	SL=600
The proposed method	PSNR	**73.61**	**71.22**	**69.65**	**68.47**	**67.52**
	STD	0.89	0.71	0.63	0.59	0.57
	SSD	**28.76**	**49.63**	**71.14**	**92.98**	**115.84**
	STD	4.33	6.20	7.88	9.48	11.56
Method in [5]	PSNR	68.09	66.38	65.00	63.84	62.92
	STD	4.60	4.46	4.40	4.33	4.29
	SSD	217.57	302.55	404.14	500.91	604.76
	STD	155.04	169.14	172.47	174.36	174.45
Method in [24]	PSNR	62.00	59.99	58.78	57.82	56.95
	STD	2.05	1.79	1.66	1.58	1.49
	SSD	456.80	710.90	931.30	1152.60	1396.80
	STD	129.86	146.56	153.21	157.00	164.92

4.3. The Overall Performance on Images

In order to evaluate the overall performance of this approach and compare it with that of the two other existing methods, all images from the BSDS500 [4] are downloaded (at June 11, 2018) and randomly generated sequences of lengths 200, 300, 400, 500, 600 are hidden into each image with the proposed approach and the other two methods. The images are stored in three different folders, including a test folder, a train folder and a val folder.

Table 6. The means and standard deviations (STD) of SSDs and PSNRs obtained with three methods on the 200 images in the test folder of the BSDS500 dataset. SL is sequence length.

Method		SL=200	SL=300	SL=400	SL=500	SL=600
The proposed method	PSNR	**73.16**	**71.28**	**69.69**	**68.51**	**67.55**
	STD	0.78	1.12	0.89	0.84	0.76
	SSD	**28.68**	**49.38**	**70.88**	**92.75**	**115.48**
	STD	4.56	6.65	8.51	10.39	12.00
Method in [5]	PSNR	67.72	65.83	64.38	63.26	62.29
	STD	4.20	4.17	4.15	4.19	4.20
	SSD	196.82	300.70	407.38	524.50	643.32
	STD	136.11	165.89	169.43	170.68	172.81
Method in [23]	PSNR	61.75	59.77	58.56	57.55	57.18
	STD	6.16	5.90	5.83	5.78	3.84
	SSD	444.40	685.60	910.30	1154.40	1396.80
	STD	129.55	144.26	154.27	160.67	166.75

Table 7. The means and standard deviations (STD) of SSDs and PSNRs obtained with three methods on the 100 images in the val folder of the BSDS500 dataset. SL is sequence length.

Method		SL=200	SL=300	SL=400	SL=500	SL=600
The proposed method	PSNR	**73.55**	**71.13**	**69.59**	**68.41**	**67.47**
	STD	0.52	0.40	0.35	0.31	0.29
	SSD	**28.94**	**50.36**	**71.62**	**94.09**	**116.66**
	STD	3.41	4.53	5.57	6.59	7.70
Method in [5]	PSNR	67.72	65.83	64.38	63.26	62.29
	STD	4.20	4.17	4.15	4.19	4.20
	SSD	196.82	300.70	407.38	524.50	643.32
	STD	136.11	165.89	169.43	170.68	172.81
Method in [23]	PSNR	61.48	60.21	59.05	58.11	57.25
	STD	7.07	2.54	2.72	2.63	2.50
	SSD	461.10	710.50	937.50	1146.80	1389.30
	STD	140.05	152.03	160.74	162.08	167.27

The test folder and the train folder both contain 200 images and the val folder contains 100 images. Tables 5, 6, and 7 show the mean values and standard deviations of the PSNRs and SSDs achieved by all methods on all sequences and images. The value of w_d is chosen to be 1 for the proposed approach. It can be seen from the tables that the mean values of PSNRs achieved by the proposed approach on images in all three folders are significantly higher than that achieved by the two other methods.

4.4. Additional Analysis and Discussions

The experimental results in subsection 4.2 show that most of stego images are generated with significantly improved similarities with their cover images by the proposed approach. The security of the proposed approach is thus higher than that of the other existing methods in general. In addition, the encryption of the information before hiding can robustly protect its content even in cases where the cover image is available to the adversary side.

From the steps of the proposed approach, it is clear that the algorithm needs $c+r$ integer keys in total to shuffle a sequence that contains up to cr bits. The key space is thus of size S^{c+r}, where S is the number of positive integers that can be determined to be co-prime with c or r by a computer. In the case where d one-dimensional logistic mappings are used for encoding, the number of possible combinations of initial values for these one-dimensional logistic mappings is at least U^d, where U is the number of real numbers that can be generated by a computer between 0 and 1.

It is straightforward to see that both S and U are generally large numbers and at least larger than 1000. The size of the key space is thus at least $10^{3(c+r+d)}$. For a sequence that contains more than 100 binary bits, if 5 one-dimensional logistic mappings are used for encoding, this number is at least 10^{75}, which suggests that the proposed approach is secure against attacks based on exhaustive search when the hidden sequence is of a moderate length.

In [14], a hybrid approach based on computing error histogram and image interpolation with greedy weights is developed for the information hiding in medical images. The proposed algorithm differs from the approach developed in [14] in two major aspects. Firstly, the proposed approach searches in the image and determine the pixels for embedding with a dynamic programming approach, while the approach in [14] performs the embedding with an adaptive image interpolation-based approach. Secondly, the proposed approach encrypts the data that needs to be hidden within an image before the embedding is performed while the approach in [14] directly embeds the data into an image.

5. Conclusions

In this paper, a new approach is proposed for image based information hiding. The hiding of a sequence of binary bits into a gray scale image can be performed in two stages. In the first stage, the binary bits in the sequence are shuffled based on a set of positive integer keys, the shuffled sequence is then encoded with a system of one-dimensional logistic mappings. In the second stage, the shuffled and encoded sequence is divided into regions of equal length and the regions are sequentially embedded into the gray values of the corresponding pixels in the given gray scale image.

A dynamic programming method is used to efficiently determine the locations of embedding that can minimize the difference between a cover image and the corresponding stego image. This approach does not need the cover image or a standard image to recover the information hidden within a stego image.

Experiments on a large number of images show that the proposed approach can achieve performance better than that of two other existing methods. The proposed approach is especially promising for perfect hiding when medical images are used as the cover images. In addition, analysis shows that the hidden information is secure against attacks based on exhaustive search. This approach is thus potentially useful for image based information hiding in a variety of applications.

References

[1] Abbasi R., Bin L., Chughtai G., Hassan H., Iqbai M., and Xu L., "A New Multilevel Reversible Bit-Planes Data Hiding Technique Based on Histogram Shifting of Efficient Compressed Domain," *Vietnam Journal of Computer Science*, vol. 5, no. 5, pp. 185-196, 2018.

[2] Abdeljawad T., Banerjee S., and Wu G., "Discrete Tempered Fractional Calculus for New Chaotic Systems with Short Memory and Image Encryption," *Optik*, vol. 218, pp. 163698, 2020.

[3] Alghafis A., Munir N., Khan M., and Hussain I., "An Encryption Scheme Based on Discrete Quantum Map and Continuous Chaotic System," *International Journal of Theoretical Physics*, vol. 59, pp. 1227-1240, 2020.

[4] Arbelaez P., Maire M., Fowlkes C., and Malik J., "Contour Detection and Hierarchical Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898-916, 2011.

[5] Chai X., Gan Z., Yuan K., Chen Y., and Liu X., "A Novel Image Encryption Scheme Based on DNA Sequence Operations and Chaotic Systems," *Neural Computing and Applications*, vol. 31, pp. 219-237, 2019.

[6] Chang C. and Tseng H., "A Steganographic Method for Digital Images Using Side Match," *Pattern Recognition Letters*, vol. 25, no. 12, pp. 1431-1437, 2004.

[7] Chen E., Min L., and Chen G., "Discrete Chaotic Systems with One-Line Equilibria and Their Application to Image Encryption," *International Journal of Bifurcation Chaos*, vol. 27, no. 3, pp. 1-17, 2017.

[8] Clark K., Vendt B., Smith K., Freymann J., Kirby J., Koppel, P., Moore S., Phillips S., Maffitt D., Pringle M., Tarbox L., and Prior F., "The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository," *Journal of Digital Imaging*, vol. 26, no. 6, pp. 1045-1057, 2013.

[9] Fan Y., Liao Y., and Li F., Zhou S., Zhang G., "Identity-Based Auditing for Shared Cloud Data with Efficient and Secure Sensitive Information Hiding," *IEEE Access*, vol. 7, pp. 114246-114260, 2019.

[10] Hamza R., "A Novel Pseudo Random Sequence Generator for Image-Cryptographic Applications," *Journal of Information Security Applications*, vol. 35, pp. 119-127, 2017.

[11] Jain A. and Rajpal N., "A Robust Image Encryption Algorithm Resistant to Attacks Using DNA and Chaotic Logistic Maps," *Multimedia Tools and Applications*, vol. 75, pp. 5455-5472, 2016.

[12] Kelkar V., Tuckley K., and Nemade H., "Novel Variants of A Histogram Shift-Based Reversible Watermarking Technique for Medical Images to Improve Hiding Capacity," *Journal of Healthcare Engineering*, vol. 2017, 2017.

[13] Khalil M., "Medical Image Steganography: Study of Medical Image Degradation when Embedding Data in the Frequency Domain," *International Journal of Computer Network Information Security*, vol. 9, no. 2, pp. 22-28, 2017.

[14] Khosravi M. and Yazdi M., "A Lossless Data Hiding Scheme for Medical Images Using A Hybrid Solution Based on Ibrw Error Histogram Computation and Quartered Interpolation with Greedy Weights," *Neural Computing and Applications*, vol. 30, no. 7, pp. 2017-2028, 2018.

[15] Li M., Lu D., Wen W., Hua R., and Zhang Y., "Cryptanalyzing A Color Image Encryption Scheme Based on Hybrid Hyper-Chaotic System And Cellular Automata," *IEEE Access*, vol. 6, pp. 47102-47111, 2018.

[16] Lin C., Chang C., and Wang, Z., "Reversible Data Hiding Scheme Using Adaptive Block Truncation Coding Based on an Edge-Based Quantization Approach," *Symmetry*, vol. 11, no. 6, pp. 765, 2019.

[17] Liu Y., Chang C., and Nguyen T., "High Capacity Turtle Shell-Based Data Hiding," *IET Image Processing*, vol. 10, no. 2, pp. 130-137, 2016.

[18] Liu Z. and Pun C., "Reversible Data-Hiding in Encrypted Images by Redundant Space Transfer," *Information Sciences*, vol. 433, pp. 188-203, 2018,

[19] Liu Y., Yang C., Sun Q., Wu S., Lin S., and Chou Y., "Enhanced Embedding Capacity for the SMSD-Based Data-Hiding Method," *Signal Processing: Image Communication*, vol. 78, pp. 216-222, 2019.

[20] Liu Y., Chang C., Huang P., and Hsu C., "Efficient Information Hiding Based on Theory of Numbers," *Symmetry*, vol. 10, no. 1, 2018.

[21] Lu T., Tseng C., Huang S., and Nhan T., "Pixel-Value-Ordering Based Reversible Information Hiding Scheme with Self-Adaptive Threshold Strategy," *Symmetry*, vol. 10, no. 12, pp. 764, 2018.

[22] Luo Y., Qin J., Xiang X., Tan Y., Liu Q., and Xiang L., "Coverless Real-Time Image Information Hiding Based on Image Block Matching and Dense Convolutional Network," *Journal of Real-Time Image Processing*, vol. 17, pp. 125-135, 2020.

[23] Song Y., Song J., and Qu J., "A Secure Image Encryption Algorithm Based on Multiple One-Dimensional Chaotic Systems," *in Proceedings of the 2nd IEEE International Conference on Computer and Communications*, Chengdu, pp. 584-587, 2016.

[24] Suthanthiramani P., Sannasy M., Sannasi G., and Arputharaj K., "Secured Data Storage and Retrieval Using Elliptic Curve Cryptography In Cloud," *The International Arab Journal of Information Technology*, vol. 18, no. 1, pp. 56-66, 2021.

[25] Teng L., Wang X., and Meng J., "A Chaotic Color Image Encryption Using Integrated Bit-Level Permutation," Multimedia Tools and Applications, vol. 77, pp. 6883-6896, 2018.

[26] Wang C., Wu N., Tsai C., and Hwang M., "A High Quality Steganographic Method with Pixel-Value Differencing and Modulus Function," *The Journal of Systems and Software*, vol. 81, no.1, pp. 150-158, 2008.

[27] Wu H., Wu N., Tsai C., and Hwang M., "Image Steganographic Scheme Based on Pixel-Value Differencing and LSB Replacement Methods," *IEE Proceedings Vision, Image and Image Signal Process*, vol. 152, no. 5, pp. 611-615, 2005.

[28] Zhang S., Gao T., and Yang L., "A Reversible Data Hiding Scheme Based on Histogram Modification in Integer DWT Domain for BTC Compressed Images," *International Journal of Network Security*, vol. 18, no. 4, pp. 718-727, 2016.

[29] Zhu S. and Zhu C., "Image Encryption Algorithm with an Avalanche Effect Based on A Six-Dimensional Discrete Chaotic System," *Multimedia Tools and Applications*, vol. 77, pp. 29119-29142, 2018.

Yinglei Song received his Ph.D. in computer science from the University of Georgia, USA in 2006. He worked as an assistant professor of computer science at the University of Maryland Eastern Shore, USA from 2007 to 2012. He is currently with the School of Electronics and Information Science at Jiangsu University of Science and Technology, China. His research interests include information hiding, algorithms for image processing, machine learning, bioinformatics and data mining.

Jia Song received her Ph.D. in computer science from Zhejiang University, China in 2014. She is currently with the Department of Electronic and Information Technology, Suzhou Vocational University, China. Her research interests include information hiding, chaotic systems, operating systems, bioinformatics algorithm design.

Junfeng Qu received his Ph.D. in computer science from the University of Georgia, USA in 2006 and joined the Department of Information Technology and Computer Science at Clayton State University as an assistant professor in that year. He is currently with the Department of Computer Science and Information Technology at the Clayton State University, USA. His research interests include information hiding, image processing, machine learning and bioinformatics.