

A New Chaotic Image Steganography Technique Based on Huffman Compression of Turkish Texts and Fractal Encryption With Post-Quantum Security

MUSTAFA CEM KASAPBAŞI^{ID}

Computer Engineering Department, Istanbul Commerce University, 34840 Istanbul, Turkey

e-mail: mckasapbasi@ticaret.edu.tr

ABSTRACT As the digital age is becoming an unavoidable part of our daily life people are becoming more concern about their privacy and security of their digital communications. Steganography and cryptography are the two most frequently used merits of digital life to fulfill these needs. In this study, a new spatial domain chaotic steganography scheme is presented utilizing also new fractal stream encryption algorithm to hide Huffman encoded and compressed Turkish texts. The study composes of four main phases. Firstly, a sample of Turkish newspaper columnist corpus is gathered to obtain not only the frequencies of letters including special Turkish characters but also punctuations, spaces, newlines, quotations, etc. As a result of the first phase, a static Huffman encoding dictionary is obtained for 102 encountered characters. Secondly, super Mandelbrot sets are used with the Logistic map to obtain one-time-pad stream keys to encrypt compressed texts. Thirdly, the LSB (Least Significant Bit) plain of the cover image is analyzed morphologically to find low entropy pixel locations so that in steganography phase spotted locations can be avoided and scheme can become more resistant against stego analysis. Lastly, the chaotic steganography phase is the final phase in which data hiding pixel is selected according to logistic map chaotic function then LSB of the selected pixel is updated. Many tests are carried out on many images to show the image quality namely PSNR, SNR, SSIM, UIQI, Entropy, MSE, and histogram. Results indicate that proposed schemes are successful for encryption and offer robust steganography.

INDEX TERMS Chaotic steganography, fractal encryption, static Huffman.

I. INTRODUCTION

Steganography is a technique consisting of a series of methodologies applied in different multimedia mediums to conceal a secret message in a carrier so that the secret message can't be recognized and noticed. Even though many secure methodologies are presented, regarding performance measures there is more space for progress in making these techniques more secure and robust. The objective of steganography is summarized in [1] as imperceptibility, security, the capacity of hiding information and robustness. However many of the studies attribute the last property to watermarking rather than steganography [1]–[3].

The associate editor coordinating the review of this manuscript and approving it for publication was Aniello Castiglione^{ID}.

Steganography techniques can be classified in two regarding the secret message embedding location, namely, spatial domain and frequency domain. However, it can be also further classified on cover image dimension, retrieval nature and on being adaptive according to carrier mediums [1].

In this study, a regional adaptive spatial domain enhanced with Fractal-Chaos stream encryption and new chaotic location selection are presented in order to contribute to the existing steganography schemes. It is regional adaptive because it utilizes well-known image processing morphology technique to obtain low entropy regions for the LSB plane of the image. These regions are avoided in the secret message-embedding phase. One of the unavoidable merits of steganography is the security of the data, in this work using a new encryption algorithm and new chaotic embedding scheme satisfies this need. The key aspects of the presented study are the Huffman

compression of Turkish texts process, the introduction of new encryption algorithm, the regional adaptive embedding location selection scheme and the chaotic pixel selection and embedding scheme.

The presented paper is organized as; in the second section, a brief literature review of the current spatial domain steganography is introduced. The third section is dedicated to describing the proposed spatial domain steganography. In results and discussion, the evaluation metrics and results are presented, comparison with the previous studies is given. Concluding remarks are given in the last section.

II. LITERATURE REVIEW

There are four important pillars that a steganography method should depend on namely: imperceptibility, secrecy, payload capacity, and robustness.

Imperceptibility means that secret message could not be understood by the human visual system or with the use of statistical techniques. Therefore many image quality metrics are used to measure imperceptibility such as histogram, Peak-Signal Noise Ratio (PSNR), MSE (Mean Squared Error), Structural Similarity Index Measure (SSIM) and other image quality indexes, etc.

If the secret message can not be disclosed and retrieved even after it is detected by statistical means then the system is considered to be secure. To improve the security of the stenographic system generally, an encryption mechanism is also offered.

Payload capacity refers to the maximum secret message that can be embedded without altering the message quality therefore imperceptibility. The embedding rate is [4] defined as the ratio of the number of hidden bits to the number of pixel in the cover image.

Robustness is determined even if the carrier image has been undergone image processing techniques such as rotation, resizing, scaling the embedded secret message can be recovered without loss of information. This property is generally required in watermarking techniques since it is mostly used for digital right management and ownership of the digital content, therefore robustness measure is not considered in this study.

In spatial domain steganography LSB, LSB matching, PVD (Pixel Value Difference), histogram shifting and pixel intensity modulation methodologies are mostly implemented examples of such studies are given respectively in [5]–[8] and [9].

The Chaos theory is applied not only in steganography but also in many encryption schemes due to its sensitivity to initial conditions [10], [11]. It is well known that in a chaotic system even a small change in initial conditions will result in totally different unpredictable outputs. This behavior is exploited either to affirm an encryption mechanism or a scrambling (diffusion) scheme. In the surveyed literature [12]–[17] mostly incorporated with the aforementioned spatial domain embedding methodologies. The results of the mentioned studies are compared and given in a later section.

In [17] for example, chaos theory is utilized to scramble the secret message then this message is embedded in the edges of the cover image. Canny edge detection algorithm is used to detect the edges. Despite the expected high payload capacity they have relatively smaller embedding capacity and low PSNR values.

In another study chaos theory used to select the embedding location of the secret message. Cover image divided into two halves upper/lower and three different chaotic maps are used to pick embedding two locations in these halves. The first 4 MSB (most significant bit) of the secret message is hidden in the last 4 bits of the selected pixel of the upper half of the cover image. The last 4 LSB of the secret message is hidden in the last 4 LSB of the selected pixel of the lower half [12].

In the study [13] a cycling chaotic function is used to generate a seed for PRNG then this PRNG is used in the select the RGB channel and the hiding position. Their algorithm has high embedding capacity and high imperceptibility results.

To the best of the author's knowledge, there is no study using chaos theory, fractal encryption and steganography in one work other than this.

III. MATERIALS AND METHODOLOGY

In this study, the offered steganography scheme composes of four distinct phases namely natural language statistic extraction and obtaining Huffman encoded output of Turkish texts; encrypting encoded output with new Fractal encryption algorithm; determining the morphologically higher entropy regions of the selected cover image; chaotic location selection for LSB steganography. The reverse operation is used for extracting the secret message.

Fig. 1 indicates the embedding procedure of the proposed scheme, which mainly includes morphological location (adaptive region) selection, fractal encryption and chaotic pixel location, and xoring procedure.

A. STATISTIC EXTRACTION AND HUFFMAN ENCODING AND COMPRESSION PHASE

In order to obtain the frequency of alphanumeric characters including punctuations in a Turkish Text, first of all, some of the Turkish Corpus' database services are explored [18], [19] however required frequency of the alphanumeric characters are not supplied or simple queries are not completed and timeout reached due to the size of corpuses. Therefore, a pilot study is carried out in order to gather a relatively small scale dataset of Turkish newspaper 14 columnists of four different mainstream newspapers. Statistics and test datasets are formed and in Table 1 general statistics of the corpus are given.

Turkish alphabet is based on the Latin alphabet and consists of 29 letters including letters special for Turkish. It is easy to find the frequencies of these letters in any Turkish corpus or even in Wikipedia [20] however, to the best of the author's knowledge, this is the first study that gives the frequency of alphabet letters with punctuations and special characters relative to each other. Table 2 gives the character

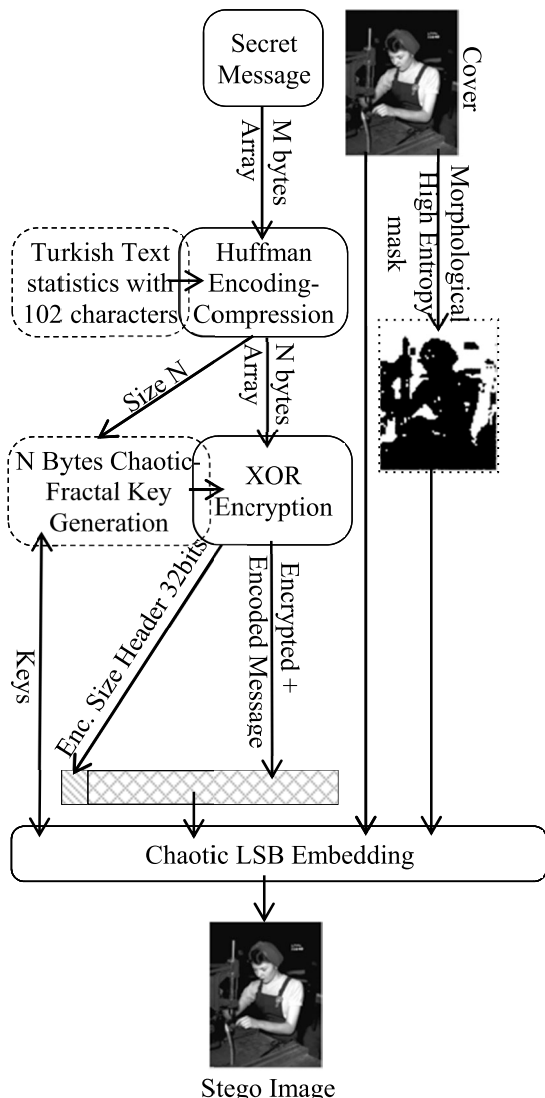


FIGURE 1. Proposed Steganography Scheme for embedding secret message.

TABLE 1. General statistics of the corpus.

	Statistical Dataset	Test Dataset
# of characters	408,062 ~600 KB	111,252 ~120KB
# of words	63,189	15,355
# of lines	5,843	2,432

and its corresponding probability and calculated Huffman Codes.

Huffman code [21] is used to find the optimum code length for the given message, which is also referred to as a lossless compression method. For every different message, a new code set is generated with a symbol dictionary. In this study, static Huffman code is used assuming that natural language texts will preserve statistical information regardless of the context with acceptable deviations. This assumption is also exploited in some cryptanalysis methods of classical encryption algorithms [22].

TABLE 2. Character probability and Huffman code.

Character	Probability	Huffman Code
#space	0,133548768	[0,1,1]
a	0,092572593	[1,1,1]
e	0,074990542	[0,0,0,1]
i	0,070364132	[0,0,1,1]
n	0,060504742	[1,0,0,0]
r	0,053781758	[1,0,1,0]
l	0,051463269	[1,1,0,0]
t	0,037102164	[0,0,1,0,1]
d	0,034035502	[0,1,0,0,1]
k	0,033870651	[0,1,0,1,0]
m	0,028538337	[1,0,0,1,1]
ç	0,026374133	[1,0,1,1,0]
y	0,026179692	[1,0,1,1,1]
u	0,025854216	[1,1,0,1,0]
s	0,024535404	[0,0,0,0,0,0]
o	0,019701872	[0,0,1,0,0,0]
b	0,017055794	[0,1,0,0,0,1]
ü	0,014517503	[1,0,0,1,0,0]
.	0,012693568	[1,1,0,1,1,1]
#enter	0,012349071	[0,0,0,0,0,1,0]
ş	0,011729821	[0,0,0,0,0,1,1]
z	0,011444501	[0,0,0,0,1,0,1]
h	0,008817444	[0,0,1,0,0,1,0]
ç	0,008642025	[0,1,0,0,0,0,0]
g	0,008631457	[0,1,0,0,0,0,1]
ğ	0,008301754	[0,1,0,1,1,0,0]
v	0,008046023	[0,1,0,1,1,1,0]
c	0,007365482	[1,0,0,1,0,1,0]
,	0,005818414	[0,0,0,0,1,0,0,0]
p	0,005759236	[0,0,0,0,1,0,0,1]
ö	0,00571908	[0,0,0,0,1,1,0,0]
A	0,004440424	[0,0,1,0,0,1,1,0]
B	0,003884579	[0,1,0,1,1,1,1,0]
f	0,003584464	[1,0,0,1,0,1,1,0]
T	0,00262283	[0,0,0,0,1,1,0,1,0]
K	0,002612263	[0,0,0,0,1,1,1,0,0]
S	0,002441071	[0,0,0,0,1,1,1,0,1]
İ	0,002432617	[0,0,0,0,1,1,1,1,0]
E	0,002111368	[0,1,0,1,1,0,1,0,0]
M	0,002024715	[0,1,0,1,1,0,1,1,0]
Y	0,001940176	[0,1,0,1,1,1,1,1,0]
D	0,00176053	[1,0,0,1,0,1,1,1,0]
0	0,001747849	[1,1,0,1,1,0,0,0,0]
H	0,001659083	[1,1,0,1,1,0,0,1,0]
1	0,00157243	[1,1,0,1,1,0,1,0,0]
N	0,001477323	[1,1,0,1,1,0,1,1,0]
R	0,001369536	[0,0,0,0,1,1,0,1,1,0]
P	0,001194117	[0,0,0,0,1,1,1,1,1,0]
O	0,001113805	[0,0,1,0,0,1,1,1,0,0]
L	0,001090556	[0,0,1,0,0,1,1,1,0,1]
-	0,001088443	[0,0,1,0,0,1,1,1,1,0]

After obtaining the Huffman codes in order to obtain further compression when necessary ZIP compression method is

TABLE 2. (Continued.) Character probability and Huffman code.

G	0,001048287	[0,1,0,1,1,0,1,0,1,1]
2	0,001003904	[0,1,0,1,1,0,1,1,1,1]
:	0,000881322	[1,0,0,1,0,1,1,1,1,0]
C	0,000817917	[1,1,0,1,1,0,0,0,1,1]
F	0,00081369	[1,1,0,1,1,0,0,1,1,0]
9	0,000805236	[1,1,0,1,1,0,0,1,1,1]
I	0,000771421	[1,1,0,1,1,0,1,0,1,1]
5	0,000722811	[1,1,0,1,1,0,1,1,1,0]
?	0,000644612	11 bits
Ç	0,000604456	11 bits
U	0,000593888	11 bits
!	0,000572753	11 bits
Ş	0,000545278	11 bits
Ö	0,000534711	11 bits
â	0,000513576	11 bits
*	0,000503009	11 bits
V	0,000494555	11 bits
3	0,000469193	11 bits
Ü	0,000469193	11 bits
"	0,000448058	11 bits
8	0,000439604	11 bits
4	0,000435377	11 bits
Z	0,00043115	11 bits
j	0,000420583	11 bits
'	0,000395221	11 bits
;	0,000262072	12 bits
6	0,000255731	12 bits
7	0,000205008	12 bits
(0,000183873	12 bits
)	0,000183873	12 bits
w	0,000171192	12 bits
J	0,000124695	13 bits
W	0,000122582	13 bits
	9,51067E-05	13 bits
Ğ	8,45393E-05	13 bits
@	6,97449E-05	14 bits
»	6,97449E-05	14 bits
/	6,76314E-05	14 bits
+	6,55179E-05	14 bits
î	6,1291E-05	14 bits
x	4,01561E-05	14 bits
û	3,59292E-05	14 bits
X	1,26809E-05	16 bits
Â	8,45393E-06	16 bits
Q	6,34044E-06	17 bits
q	6,34044E-06	17 bits
%	2,11348E-06	18 bits
&	2,11348E-06	18 bits
é	2,11348E-06	19 bits
á	2,11348E-06	19 bits
ó	2,11348E-06	18 bits

also preferred. Then the compressed message is transferred to the fractal encryption phase. As can be seen from Table 3 with

TABLE 3. Compression comparisons.

Not encoded size in bits	Huffman bits	Zip bits	Huffman+Zip bits
8192	4840	9990	4840
40960	23944	25888	24624
81920	47576	39720	45672

the small size of texts better compression results are obtained with proposed Huffman encoding relatively to the ZIP.

B. FRACTAL ENCRYPTION PHASE

In order to make the message more secure to active and passive attacks, a new fractal and chaotic blended encryption phase are introduced. This encryption scheme works as if stream cipher and one-time pad. It has a unique key generation algorithm that allows producing the same unique key size with the payload size. Aftermath XOR operation is performed between the generated key and the payload. Separately, the size value of the message is encrypted to form a 32-bit-header, which is appended to the beginning of the encrypted message.

The core part of the proposed encryption method is to generate the same size key with the message it gives stream and one-time pad cipher like behavior the encryption. Keys are obtained using super Mandelbrot fractals and Logistic map chaotic function.

In the following subsection, Mandelbrot fractals will be described briefly and since logistic map chaotic function is also used in the message hiding phase it will be mentioned in that section.

1) MANDELBROT SETS

Mandelbrot [23] while analyzing the Julia set with different parameters for connectivity property discovered new fractal sets named after him also called M-set. Mandelbrot fractal set has been extensively used in many different areas recent examples of which can be found in [24], [25].

Using c as a complex parameter and z as a complex function, Mandelbrot set utilizes the complex function given in (1). These sets have been demonstrated for higher degree polynomials like quadratic, cubic and so on. Fig. 2 illustrates the Mandelbrot sets for $n=2$ to $n=5$.

$$z_{i+1} = z_i^n + c$$

$$c = t + yi \tag{1}$$

In this study, cubic polynomial iteration is used and initial values of z and c are considered as a key tuple for the encryption process. To increase the keyspace exponent of the z can also be assumed as a part of the encryption key tuple.

As key tuple for fractal (y_1, t_1, y_2, t_2, n) is used and a key tuple for the chaotic logistic map is (λ, x_0) . When determining the key values below ranges are selected in regards to previous

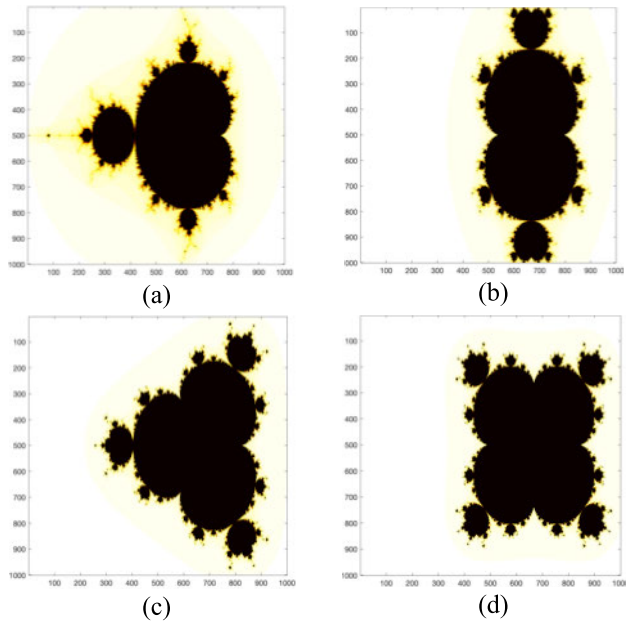


FIGURE 2. Mandelbrot sets for $n=2, 3, 4$ and 5 in subfigures (a), (b), (c), and (d) respectively.

studies:

$$y = \{y | y \in \mathbb{R} \text{ y is in } [-1.4, 1.1]\},$$

$$t = \{t | t \in \mathbb{R} \text{ t is in } [-2, 1]\},$$

$$n = \{n | n \in \mathbb{Z} \text{ n is in } [2, 10]\}.$$

2) PROPOSED FRACAL ENCRYPTION METOD

In this study, in order to generate the encryption key with the size of the payload, two different Mandelbrot sets, and a Logistic Map chaotic function is exploited. Then the generated key is used to perform XOR operation. The detailed algorithm of the encryption system is given in Algorithm 1.

In the presented study the values of the keys are given as below so that anyone can repeat the results: $\lambda = 3.991461146114611$, $x_0 = 0.146114611461146$, $y_1 = -0.54610000000000$, $y_2 = .2111119000000000$, $t_1 = .346100000000000$, $t_2 = -0.331111900000000$, $n = 3$.

3) BRIEF SECURITY ANALYSIS OF PROPOSED ENCRYPTION SYSTEM

Since the encrypted output is not hidden as a block and is used incorporation with the chaotic embedding algorithm to be diffused to suitable pixels of the cover image. Moreover, for every different size of the message, a new different key is generated, which gives an extra layer of protection. Therefore the key size of the encrypted message (keyspace) will be discussed in this subsection.

All the keys used in the double precision and therefore they have 15 significant digit levels. y_1, y_2, t_1 and t_2 are used in fractal part and λ, x_0 are used in chaotic part as key. For y_1, y_2, t_1, t_2 and λ only last 13 least significant decimal part used as key for x_0 last 15 least significant decimal part used as

Algorithm 1 Encrypting the Encoded-Compressed Secret Message or Message Size Value to 32 bit

Input: Encoded-compressed Message bit array as M , Message Size value as MS , Fractal key quintuple (y_1, y_2, t_1, t_2, n) , Logistic map key tuple (λ, x_0) , i represents imaginary part of the complex number

Output: Encrypted Header + Encrypted Enc.

Begin

```

    HS ← Size(M);
    MS[] ← ConvertBitArray(MS)
    c1 ← t1 + y1 i
    z1[] ← Array [13]
    c2 ← t2 + y2 i
    z2[] ← Array [13]
    x1 ← x0
    for i=1, 2 ..., 31 do
        z1[i+1] ← z1[i]^n + c1
        z2[i+1] ← z2[i]^(n+1) + c1
    endfor
    for i=1, 2 ..., 32 do
        dif1 ← 100*Abs(z1[i]- z2[i])
        keyBit ← Floor(Mod(dif1,2))
        key1[i] ← keyBit
        x1 ← λx1(1 - x1)
        dif2 ← Floor(x1*10^15)
        key2[i] ← Mod(dif2,2)
        key3[i] ← Xor(key1[i], key2[i])
        MS[i] ← Xor(key3[i], MS[i])
    endfor
    for i=1, 2 ..., HS do
        x1 ← λx1(1 - x1)
    endfor
    c1 ← t1 + y1 i
    z1[] ← Array[HS]
    c2 ← t2 + y2 i
    z2[] ← Array[HS]
    for i=1, 2 ..., HS-1 do
        z1[i+1] ← z1[i]^n + c1
        z2[i+1] ← z2[i]^(n+1) + c1
    endfor
    for i=1, 2 ..., HS do
        dif1 ← 100*Abs(z1[i]- z2[i])
        keyBit ← Floor(Mod(dif1,2))
        key1[i] ← keyBit
        x1 ← λx1(1 - x1)
        dif2 ← Floor(x1*10^15)
        key2[i] ← Mod(dif2,2)
        key3[i] ← Xor(key1[i], key2[i])
        M[i] ← Xor(key3[i], M[i])
    endfor
    return Concatenate(MS, M)
end

```

key. n is a small integer number therefore keyspace of this key is ignored in calculations. The total key size becomes

$10^{80} \approx 2^{266}$ which is indicating very large keyspace, which is secure enough for both conventional computing systems and post-quantum systems according to [26]. Moreover, since there is a chaotic pixel location selection process and the keys of the process can be also independent of the encryption process therefore it increases keyspace 10^{28} times more almost $\approx 2^{93}$ times. Therefore total key searching space becomes almost $\approx 2^{359}$.

The cryptography systems are expected to confuse and diffuse the plain text and produce almost uniformly distributed outputs. When such outputs are hidden in an LSB plane of the cover image, a stego analyst will first investigate this plane for such uniform information, because ordinary images generally are not uniformly distributed [27].

Therefore encrypted messages should not be embedded as block in Stego; every bit of message should be hidden as sparse as possible. In order to prove this intuitive hypothesis chaotic location selection procedure is implemented which ensures that every consecutive bit is placed in the uncorrelated location of the LSB plane. Then chi-square tests are performed. Since the encrypted output is scattered all over the LSB plane using chaotic algorithm it is not needed to check the randomness of the output. When the message has linearly embedded the presence of the secret message is discovered, however, it is not discovered with the proposed system.

Since it does not have substitution and permutation rounds the proposed encryption method will not produce perfect uniformly distributed outputs, which results in less detectable Stego images quantitative results of which are given in later sections.

C. DETERMINING THE MORPHOLOGY OF THE LSB PLANE

Determining the morphology of the LSB plane of the cover image is important because the hiding process is going to take place in that (those) plane(s). It is desired that the plane should have a maximum capacity for hiding places, in other words, it should not possess low entropy regions and if such regions exist those regions should be avoided for data hiding.

In this study after the LSB plane is sliced from the cover image, this plane is subjected to some series of Morphological operations with respect to the selected structural element. The operations used in the study are called morphological closing and opening operations, which are just applying dilation and erosion fundamental morphological operations in a different sequence. The definition of the mathematical morphological is based on the set theory. Providing that $A \subseteq \mathbb{Z}^2$ is a binary image and the kernel $B \subseteq \mathbb{Z}^2$ then some of the morphological operations can be expressed as given from (1) to (7), namely reflection, p transition, erosion, dilation, opening and, closing respectively.

$$\hat{A} = \{w|w = -a, \forall a \in A\} \tag{2}$$

$$(A)_p = \{w|w = a + p, \forall a \in A\} \tag{3}$$

$$A \ominus B = \{z|(B)_z \subseteq A\} \tag{4}$$

$$A \oplus B = \{z|(\hat{B})_z \cap A \neq \emptyset\} \tag{5}$$

$$A \circ B = (A \ominus B) \oplus B \tag{6}$$

$$A \cdot B = (A \oplus B) \ominus B \tag{7}$$

Since these operations are fundamental for image processing readers may want to consult [28] for detailed descriptions. However, the algorithm for determining the low entropy locations is given in Algorithm 2 which contains some minor changes from the algorithm given in [29].

Since the LSB plane is a bit plane, determining the entropy of the LSB plane will give 1 for the highest entropy when calculated from (8). Therefore an entropy threshold of 0.9999 is selected to if the LSB plane has a high entropy level. The Same equation is used to assess the entropy of the cover and LSB bit plane.

$$H = - \sum_{i=0}^n p(x_i) \log_2 p(x_i) \tag{8}$$

In (8) entropy is used to show the randomness of the digital image, $p(x_i)$ is the probability of random variable x (in our case 0 or 1) at i th index. n is the maximum value of the pixel which is 255. In images, a low entropy region indicates similar pixel values where any small change can be observed even with the human eye as depicted in Fig. 3 (f). Therefore such regions must be avoided from message embedding.

Algorithm 2 Determining Usable Regions of LSB Plane

Input: Cover image (color or grayscale image) **C**

Output: Locations of high entropy regions in 1D array

Begin

height \leftarrow **Height**(C)

width \leftarrow **Width**(C)

SE \leftarrow **StructuralElement**(‘disk’, 7×7)

LP \leftarrow second last plane of the GrayScale (C)

LP_Entropy \leftarrow **Entropy**(LP)

if(LP_Entropy > 0.9999)

for $i=0, 1, 2, \dots, \text{height} * \text{width}$ **do**

 locations(i) \leftarrow i

end for

return locations

endif

LO \leftarrow **ImageOpening**(LP, SE)

LC \leftarrow **ImageClosing**(LP, SE)

OUT \leftarrow **or**(LO, **not**(LC))

LO \leftarrow **ImageOpen**(LP, SE)

$k \leftarrow 1$

for $i=0, 1, 2, \dots, \text{height} * \text{width}$ **do**

if (LO(i) == 0)

 locations(k) \leftarrow i

$k \leftarrow k+1$

endif

end for

return locations

end

In Fig 3(a) a sample image with low LSB plane entropy is given. Fig. 3(b) depicts the sliced second LSB plane of

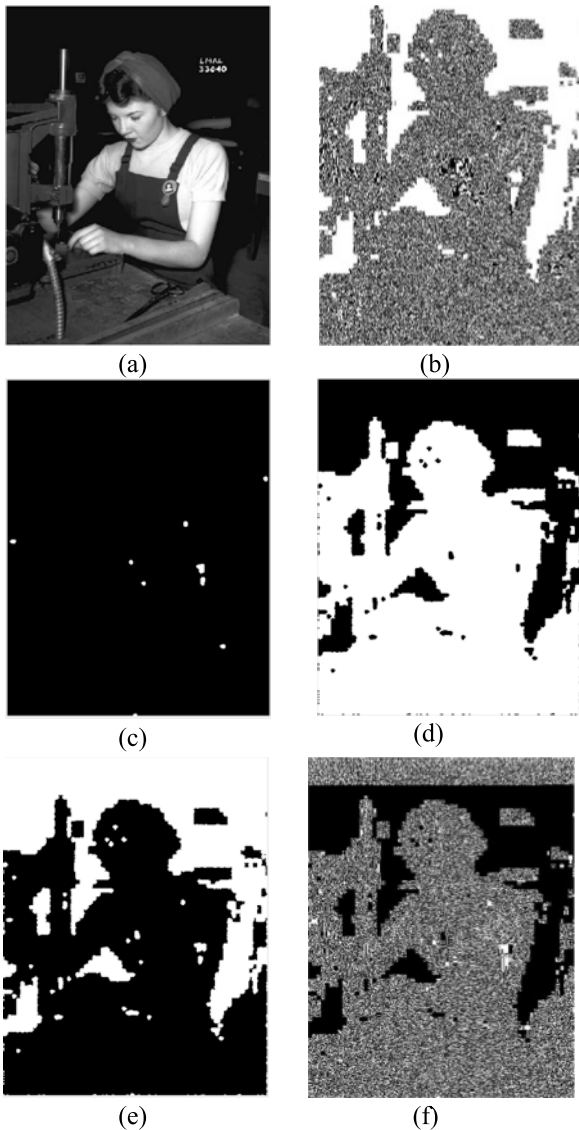


FIGURE 3. (a) Original image, (b) Second LSB plane of the image, (c) Morphological opening operation of the LSB plane, (d) Morphological closing operation of the LSB plane, (e) High entropy region mask obtained from logical or operation (d) and the inverse of Fig. 3(c) and the result is depicted in Fig. 3(e). Fig. 3(f) is a stego image in which the 5KB secret message is linearly embedded, despite the relatively small amount of payload and stego image has a PSNR value as high as 61.749401263, the existence of the secret message becomes perceptible even with the human visual system. Therefore before one should consider the morphological status of the LSB plane in carrier image before embedding even small amount of payload. In this study second, LSB bit plane is used to extract morphology since the LSB plane will be used for

the image. As it consists of a relatively large area of consecutive 0s and 1s, morphological opening and closing operations will give these regions with respect to structural element, which is 7×7 disk kernel of 1s. The high entropy region is obtained after getting the logical or operation of Fig. 3(c) and the inverse of Fig. 3(d) and the result is depicted in Fig. 3(e).

Fig. 3(f) is a stego image in which the 5KB secret message is linearly embedded, despite the relatively small amount of payload and stego image has a PSNR value as high as 61.749401263, the existence of the secret message becomes perceptible even with the human visual system. Therefore before one should consider the morphological status of the LSB plane in carrier image before embedding even small amount of payload. In this study second, LSB bit plane is used to extract morphology since the LSB plane will be used for

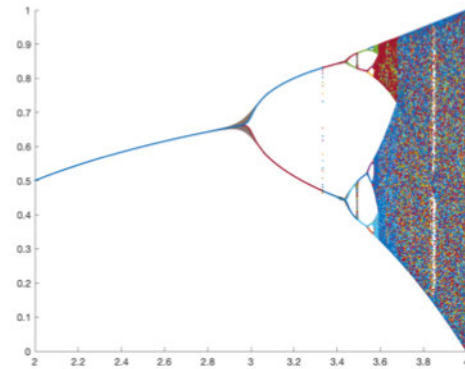


FIGURE 4. Bifurcation diagram of logistic map chaotic function with respect to different values of λ (horizontal axis) = 2 to 4 with spacing 0.0001 and $x_0 = 0.1$.

embedding the secret message, on the receiver side it would not be possible.

D. DETERMINING CHAOTIC LOCATIONS FOR STEGANOGRAPHY

Many Chaotic algorithms and schemes are used in different spatial domain steganography applications. Chaotic dynamics are well known for their impromptu behavior results from some nonlinear dynamical systems. Therefore they are used as a source of diffusion in security techniques [30].

In [31] chaotic is used not used for selecting the hiding location but to scramble the information before hiding. In the [13] chaotic scheme is used to generate a seed for PRNG which will be effective in selecting color channels and pixels. A similar approach in [32], represented accompanying industry standard encryption schemes however they have not mentioned the morphology of the cover image. [12] utilizes three different chaotic maps to select a channel of pixel and, coordination of pixel with respect to height and width for hiding the secret message, however, this method does not offer cryptography or morphological analysis of used LSB planes.

In the presented study a very well known chaotic function named logistic map is used to select a pixel location for hiding secret data. Logistic map in its first applications is used to model for population growth or rate of heating the continuous differential equation of which is given in (9) and discrete form of the equation can be represented as in (10):

$$\frac{dx}{dt} = \lambda x(1 - x) \tag{9}$$

$$x_{i+1} = \lambda x_i(1 - x_i) \tag{10}$$

Initial parameters x_0 and λ can be in $x_0 \in (0, 1)$ and $\lambda \in (0, 4)$. The chaotic character of the equation highly depends on the selected initial value of λ . When $\lambda \in [3.64, 4)$ is selected in the iterative sequence and model shows chaotic behavior [33]. The bifurcation diagram of the logistic map is given in Fig. 4 to better visualize such behavior.

In Algorithm 3 it is intended to find a pixel location relative to its width and height using the chaotic function. As inputs

Algorithm 3 Determining Pixel Location Chaotic Based

Input: width **W**, height **H** of Cover, Locations List **L**(from output of Algorithm 1), λ , x_i
Output: Updated Location List **L**, Coordinates of the pixel location (**w**, **h**), x_{i+1}
begin
 size \leftarrow Size(L)
 $x_{i+1} \leftarrow \lambda x_i(1 - x_i)$
 pixelLocationIndex $\leftarrow x_{i+1} * \text{size}$
 linearLocation \leftarrow L(pixelLocationIndex)
 if (mod(linearLocation, W) \neq 0)
 w \leftarrow W
 h \leftarrow int(linearLocation/W)
 else
 h \leftarrow int(linearLocation/W)+1
 w \leftarrow mod(linearLocation, W)
 endif
 L(pixelLocationIndex).remove
end

it requires quintuple (W, H, Locations[], x_i , λ). W, H are the width and height of the cover image respectively, Locations[] array is the output of the Algorithm 2 which is mainly high entropy regions of the cover image. x_i is the previously calculated chaotic function output, λ is the coefficient of the logistic map chaotic map.

IV. RESULTS AND DISCUSSIONS**A. PERFORMANCE EVALUATION TECHNIQUES**

1) PAYLOAD CAPACITY (BPP)

It is the measure used to describe how much message is embedded in the cover image generally interpreted in terms of Bit per Pixel bpp and calculated as given. This metric can be calculated as in (11)

$$\text{Capacity}(bpp) = \frac{\# \text{ of embedded secret bits}}{\text{Total pixel in cover image}} \quad (11)$$

2) PEAK SIGNAL NOISE RATIO (PSNR)

As the pixels of the cover image are changed the quality of the image is going to change which may have a dramatic influence on imperceptibility. PSNR is used to analyze the quality of the stego-image by calculating the mean squared error (MSE) value between the stego-image and the cover. For grayscale images, PSNR is calculated as (12) [1]:

$$\text{PSNR (dB)} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right)$$

$$\text{MSE} = \frac{\sum_{i=1}^N (C_i - S_i)^2}{N} \quad (12)$$

N is the number of the pixel C_i and S_i are the i th pixel value of cover and stego-image respectively.

3) STRUCTURAL SIMILARITY INDEX MEASURE (SSIM)

The Structural Similarity (SSIM) is used to assess quality depending on a multiplicative combination of three terms,

namely, the structural term, the luminance term, and the contrast term [34] in (13):

$$\text{SSIM}(x, y) = (s(x, y))^\alpha (l(x, y))^\beta (c(x, y))^\gamma \quad (13)$$

where

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

in these equations x is represents a set of pixel values of the image, y is the set of corresponding pixel values of the compared image. μ_x , μ_y , σ_x , σ_y and σ_{xy} are means, standard deviation, cross-covariance for image x and y respectively. If the exponents (α , β and γ) and coefficients (C_1 , C_2 and C_3) are set to default values then the equation becomes as given in (14).

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (14)$$

4) UNIVERSAL IMAGE QUALITY INDEX (UIQI)

In order to overcome some questionable properties of PSNR and MSE Universal Image quality index is offered by [35]. The index value ranges $[-1, 1]$ and 1 indicating the highest quality achieved when images are the same. This quality index tries to model any distortion as a result of three different factors namely correlation, luminance and contrast. The equation of the UIQI can be given as in (15),

$$\text{UIQI}(x, y) = \frac{\sigma_{xy}}{\sigma_x\sigma_y} \frac{2\mu_x\mu_y}{\mu_x^2 + \mu_y^2} \frac{2\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2} \quad (15)$$

5) CHI-SQUARE TEST FOR STEGO ANALYSIS

Chi-Square test is a statistical test first introduced in [27] to discover the probability of the presence of a hidden message in an image. The main assumption of the test is that for ordinary images the LSB plane is not completely random, and frequencies of pair of values (PoVs) tend to be far when no message embedded and tend to be equal when uniformly distributed message is embedded. In this study, Chi-Square test implemented in [36] is utilized.

The theoretically expected frequency distribution in steganogramme is compared with some sample distribution of stego. Following (16) is used to get the chi-square statistic of frequency differences between PoVs:

$$X_{n-1}^2 = \sum_{i=0}^{127} \frac{(x_i - \mu_i)^2}{\mu_i}, \quad \text{where } \mu_i = \frac{x_i + y_i}{2} \quad (16)$$

x and y are set of PoV frequencies and n indicates the degree of freedom. The last step is calculating the probability of embedding, by integrating the density function with X_{n-1}^2 as

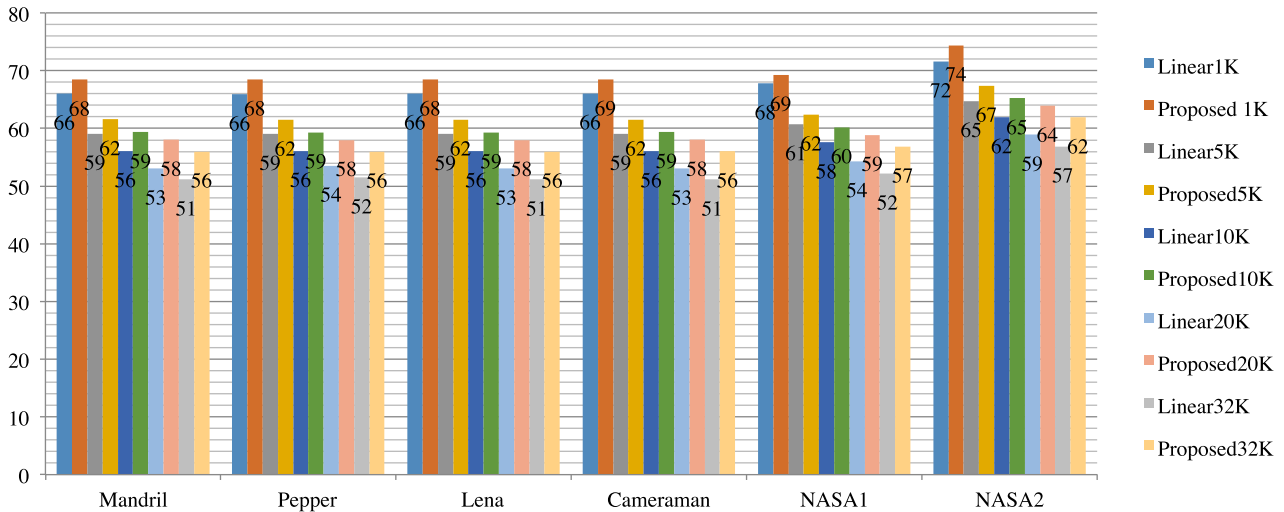


FIGURE 5. PSNR comparisons of methods for different images with different payload sizes.

its upper limit as given in (17):

$$p = 1 - \frac{1}{2^{\frac{n-1}{2}} \Gamma(\frac{n-1}{2})} \int_0^{x^2} e^{-\frac{u}{2}} u^{\frac{n-1}{2}-1} du \quad (17)$$

where Γ is the Euler Gamma function [3].







B. RESULTS AND COMPARISONS

In order to assess the performance of the proposed scheme, a series of tests are carried out with different carrier images and with different payload sizes. Test images are chosen between well-known images namely Mandril, Pepper, Cameraman, and Lena obtained from [37] SIPI image database, some of the less well-known images are obtained from NASA image and video repository [38]. Message to be embedded is selected not in the train set of the Huffman Encoding phase. To compare the performance metrics Linear LSB, Random LSB and the Proposed method are compared. Table 4 summarizes the image properties used in the tests. Sizes of the embedded secret messages range from 1KB to 93.5KB, which is empirically obtained maximum payload size for 512×512 cover image. As mentioned before only the LSB of the selected cover image is changed.

Fig. 5 indicates the PSNR value comparison of the proposed system with the linear embedding system. As it can be interpreted from the figure PSNR values are better for every different payload sizes ranging from 1K to 32K. After 32K payload size, it is not possible to linearly embed a larger size of secret message inside a 512×512 image since only one bit change is made (1 BPP). However, with the proposed method it is possible to obtain a BPP value as high as 2.93.

Table 5 is given to indicate detailed performance metrics of the given system with PSNR, SNR, Entropy, SSIM, MSE and UIQI. As it can be seen from the table proposed system has high results. However in images where last LSB plane entropy are not suitable as in PEPPER, NASA1 and NASA2, despite enough amount of pixel available, system does not

TABLE 4. Properties of the test images *Last Plane Morphology (LPM). *Also known as Baboon.

Image	Size	Original Entropy	LSB-Plane Entropy	LPM*
 Mandril	512x512	7.292548775	0.999999805847702	OK
 Pepper	512x512	6.762425168	0.999521962	NO
 Lena	512x512	7.445061353	0.9999942207286	OK
 Cam.	512x512	7.047955232	0.999999644613946	OK
 NASA1	640x499	5,612678017	0.911676208636996	NO
 NASA2	640x531	4.4208779771	0.999234737106807	NO

allow embedding since the morphology of the last plane is not suitable. If it was embedded to those regions it would be visible with even human visual system.

From Fig. 6 to Fig. 12 original cover images and corresponding histograms, also stego images and corresponding histograms are given for mentioned payloads. As can be concluded from these figures there is no noticeable shift in the histograms when compared with the original ones.

TABLE 5. Comparison table of performance metrics.

		32K Payload- 1 BPP for 512*512					
Pixel Selection	Image	PSNR	SNR	Entropy	SSIM	MSE	UIQI
Linear	Mandril	51.15401	45.60023	7.27047	0.99837	0.49852	0.99768
	Pepper	51.50817	45.56559	7.23804	0.99733	0.45948	0.99307
	Lena	51.13528	45.47878	7.42366	0.99605	0.50067	0.98611
	Cameraman	51.13266	45.51622	7.03077	0.99502	0.50097	0.94449
	NASA1	52.17287	42.89640	5.89452	0.99673	0.39427	0.68615
	NASA2	56.88314	45.15547	4.45305	0.99787	0.13328	0.79268
Proposed	Mandril	56.00248	50.44870	7.29259	0.99946	0.16324	0.99926
	Pepper	55.98374	50.04116	7.05906	0.99896	0.16395	0.99753
	Lena	55.98758	50.33109	7.44557	0.99862	0.16380	0.99466
	Cameraman	56.01823	50.40180	7.05046	0.99832	0.16265	0.98048
	NASA1	56.85866	47.58220	5.62482	0.99894	0.13403	0.99151
	NASA2	61.89992	50.17225	4.42942	0.99942	0.04198	0.98115
Pixel Selection	Image	48K Payload 1.5 BPP for 512*512					
Proposed	Mandril	54.21690	48.66313	7.29278	0.99920	0.24626	0.99890
	Pepper	54.23551	48.29293	7.13941	0.99849	0.24520	0.99636
	Lena	54.22519	48.56869	7.44572	0.99793	0.24579	0.99214
	Cameraman	54.21677	48.60033	7.05168	0.99746	0.24627	0.97063
	NASA1	55.09880	45.82233	5.63009	0.99845	0.20100	0.98863
	NASA2	Morphology test rejects secret message embedding that large					
Pixel Selection	Image	64K Payload 2 BPP for 512*512					
Proposed	Mandril	52.83736	47.28358	7.29275	0.99891	0.33833	0.99849
	Pepper	52.83858	46.89600	7.20110	0.99798	0.33824	0.99505
	Lena	52.83325	47.17675	7.44587	0.99720	0.33865	0.98960
	Cameraman	52.82108	47.20464	7.05237	0.99650	0.33960	0.96030
	NASA1	53.70477	44.42830	5.63496	0.99792	0.27708	0.98592
	NASA2	Morphology test rejects secret message embedding that large					
Pixel Selection	Image	90K Payload 2.5 BPP for 512*512					
Proposed	Mandril	51.94437	46.39059	7.29286	0.99867	0.41557	0.99815
	Pepper	51.94664	46.00406	7.23080	0.99759	0.41535	0.99410
	Lena	51.93521	46.27872	7.44598	0.99660	0.41645	0.98769
	Cameraman	51.95343	46.33699	7.05285	0.99576	0.41470	0.95225
	NASA1	Morphology of cover image is not suitable for such large payload					
	NASA2	Morphology of cover image is not suitable for such large payload					
Pixel Selection	Image	93,5K Payload 2.93 BPP (Maximum) for 512*512					
Proposed	Mandril	51.15378	45.60001	7.292826	0.99836	0.498543	0.99766
	Pepper	Morphology of cover image is not suitable for such large payload					
	Lena	51.14419	45.48770	7.44599	0.99599	0.49965	0.985935
	Cameraman	51.15285	45.53641	7.05301	0.99497	0.49865	0.943944
	NASA1	Morphology of cover image is not suitable for such large payload					
	NASA2	Morphology of cover image is not suitable for such large payload					

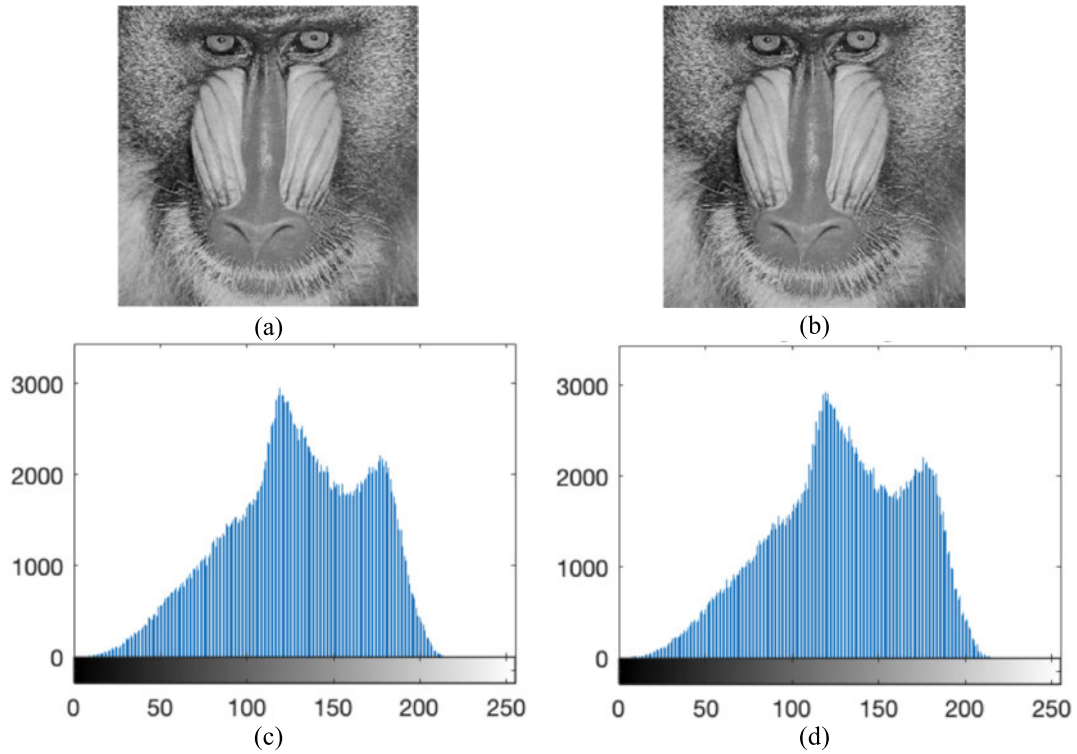


FIGURE 6. Original image (a), stego image (b), original histogram (c) and stego histogram (d) of Mandril image with 93,5 KB secret message.

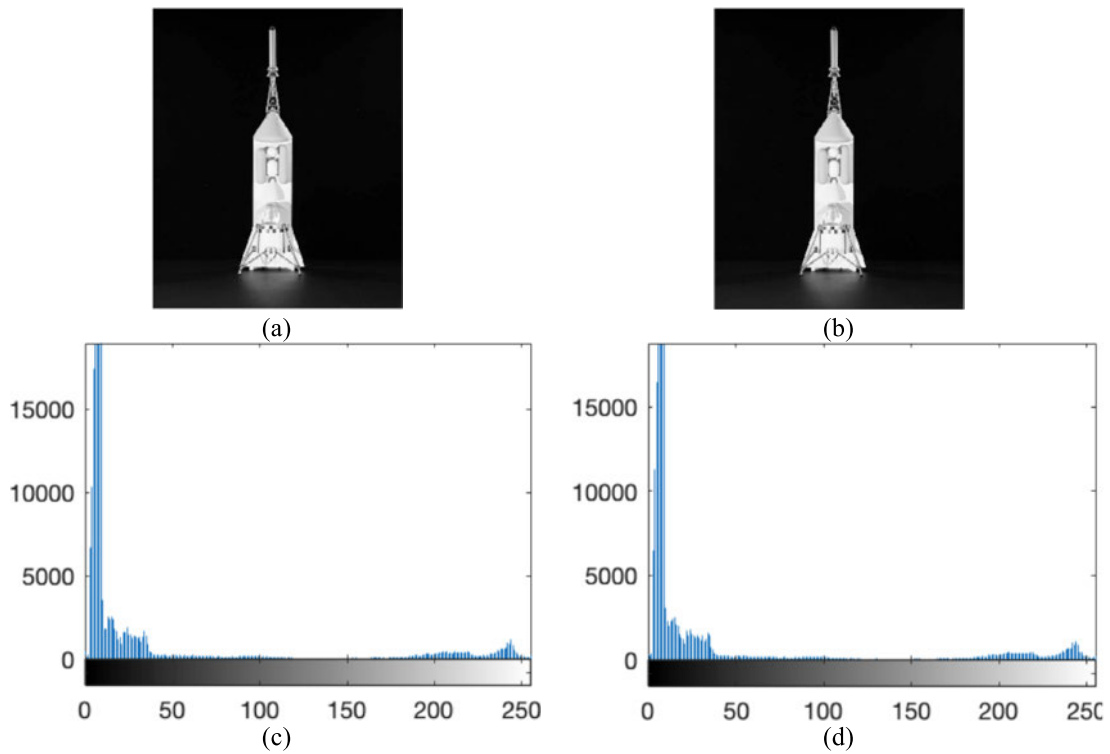


FIGURE 7. Original image (a), stego image (b), original histogram (c) and stego histogram (d) of NASA2 image with 32 KB secret message.

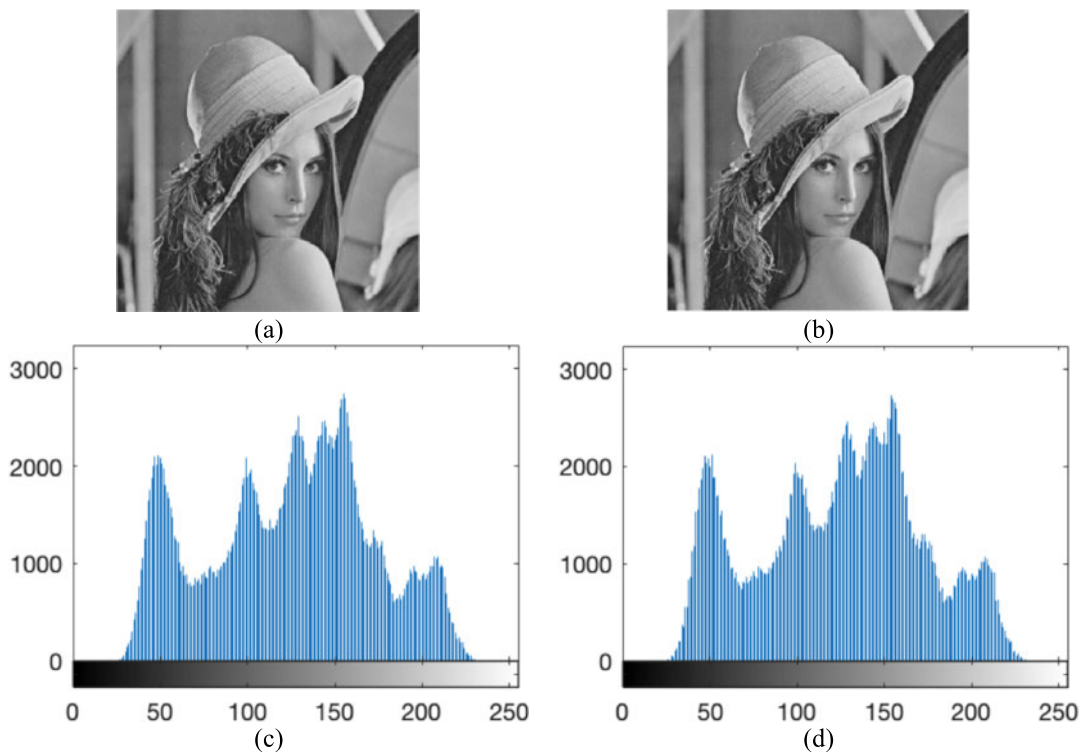


FIGURE 8. Original image (a), stego image (b), original histogram (c) and stego histogram (d) of LENA image with 93,5 KB secret message.

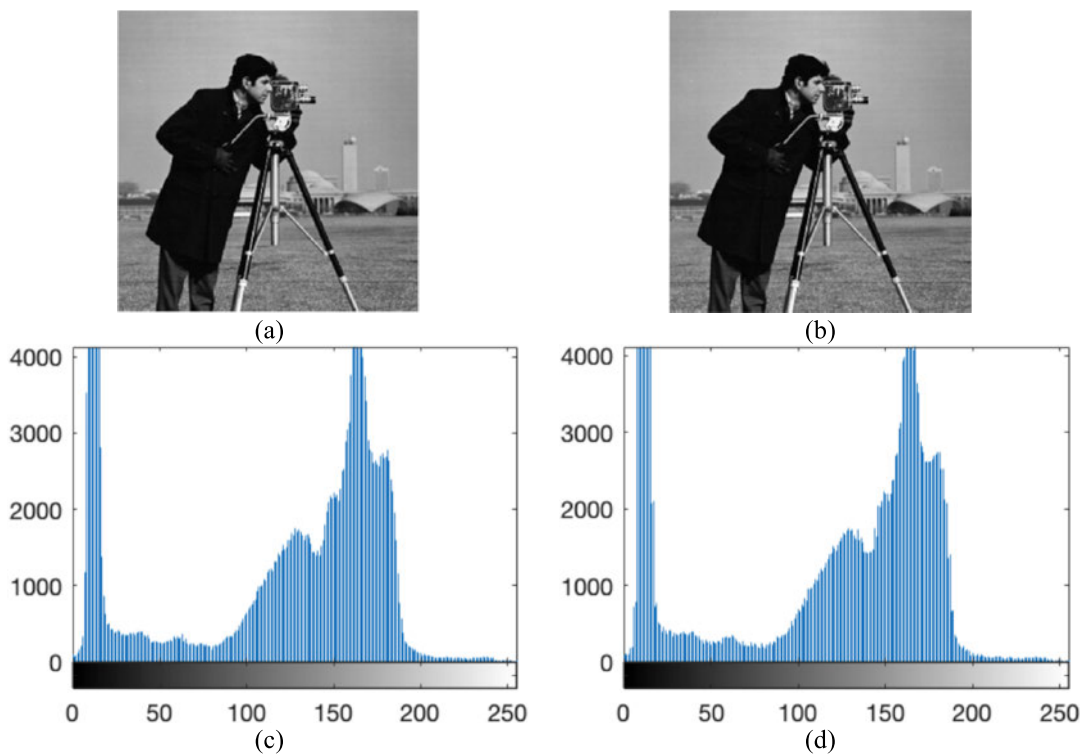


FIGURE 9. Original image (a), stego image (b), original histogram (c), and stego histogram (d) of cameraman image with 93,5 KB secret message.

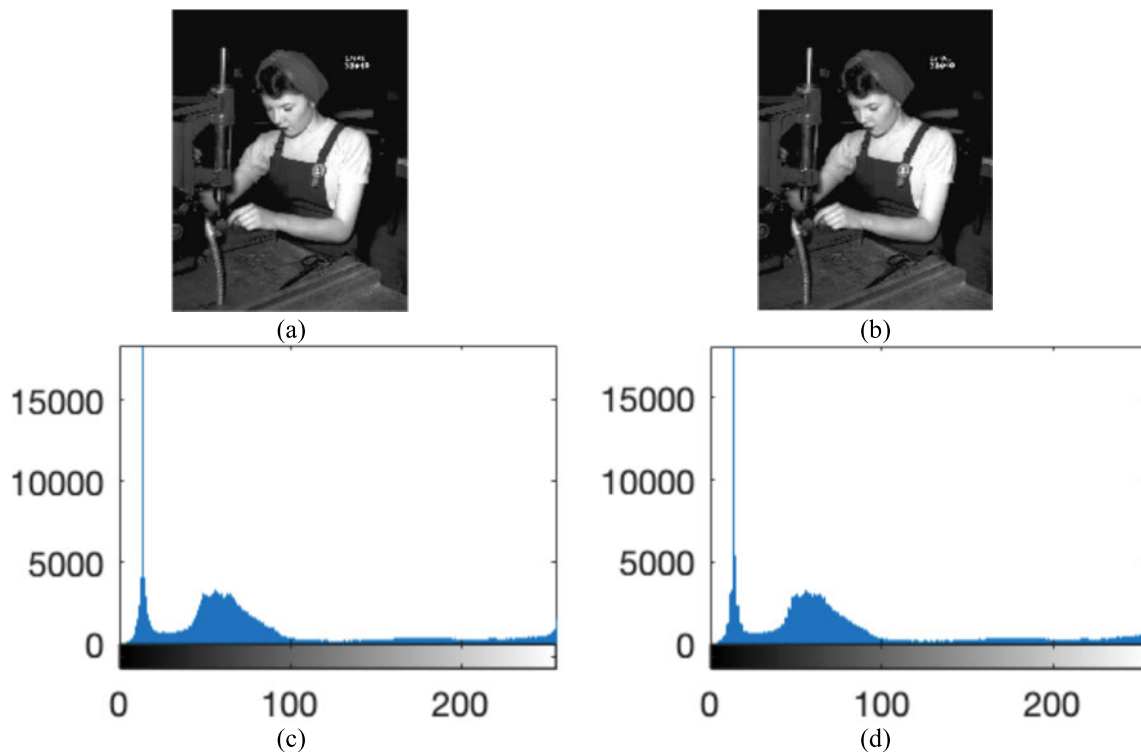


FIGURE 10. Original image (a), stego image (b), original histogram (c), and stego histogram (d) of NASA1 image with 64 KB secret message.

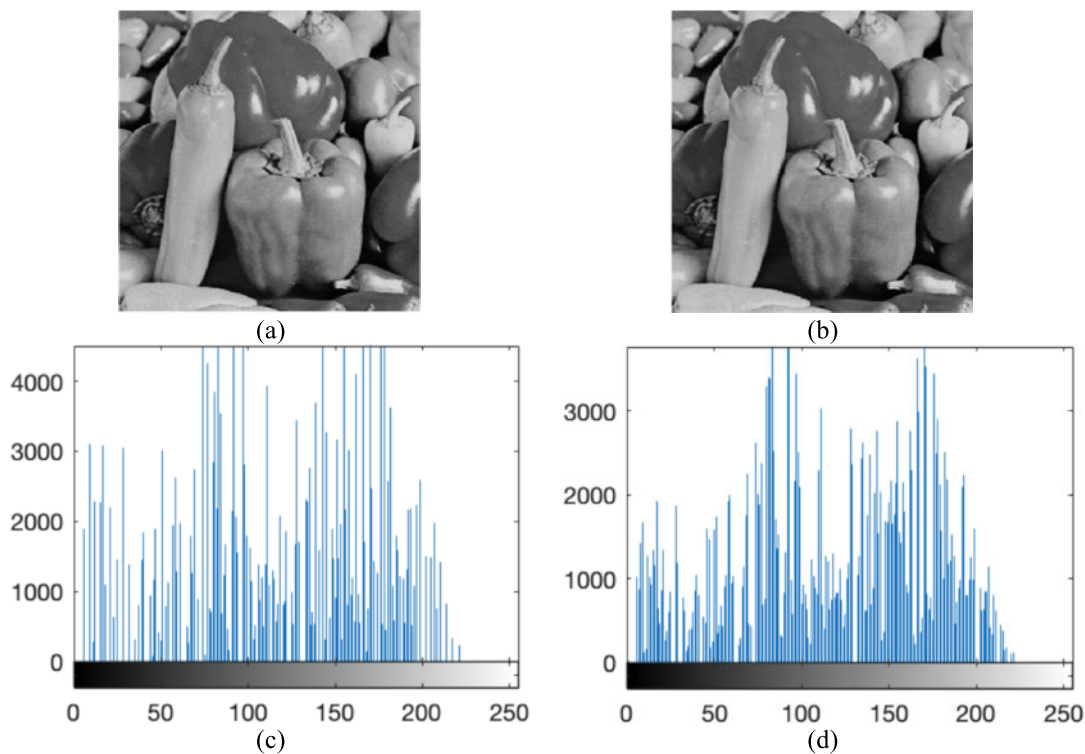


FIGURE 11. Original image (a), stego image (b), original histogram (c), and stego histogram (d) of pepper image with 90KB secret message.

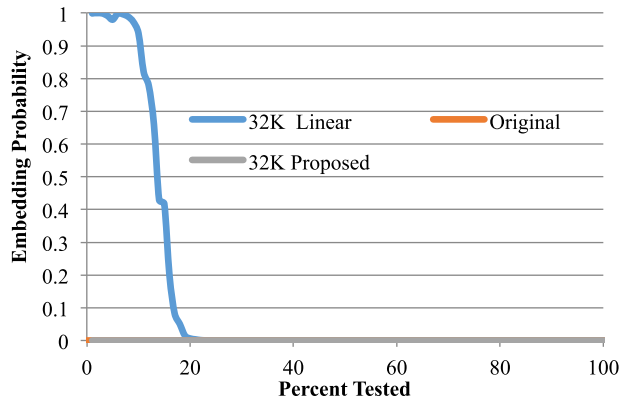


FIGURE 12. Chi-Square test results for original, linear 32KB embedding, proposed method 32KB embedding.

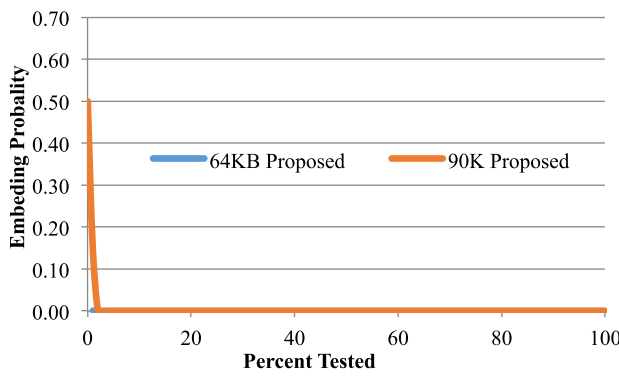


FIGURE 13. Chi-Square test results for proposed method with 64KB and 90KB Embedding.

1) CHI SQUARE TEST FOR STEGO ANALYSIS

The chi-square attack is used to assess the hiding performance analysis of the proposed system. In order to better compare the results of the chi-square tests, the original pepper image without embedding, 32 KB payload linearly embedded, and 32 KB payload embedded by the proposed methodology is given in Fig. 12. As can be inferred from the figure the test reveals the presence of the secret message for linearly embedding. However it fails to discover the existence of the secret message with the same payload, which is almost the same as the original image test result.

In Fig. 13 indicates results for larger payload in the same image cover image. Results indicate that even in larger payloads like 64KB and 90 KB the proposed system passes the Chi-Square test. In Fig. 13 the plot of proposed method embedding probability is very close to zero only small portion of it visible under the other plot.

2) POST QUANTUM ANALYSIS OF THE PROPOSED SYSTEM

Post Quantum systems are claimed to be the systems that are going to still fulfill the security requirements not only for the conventional computing environment but also for the quantum computing environment [26].

The proposed system is considered to remain secure in the post-quantum era for such reasons; the security of the

TABLE 6. Comparison table of performance metrics.

Related Studies	Method	PSNR	Capacity
[14]	LSB + Chaotic	44.53	2 BPP
[10]	LSB + Chaotic (RGB cover)	55.4126	(4 bits per 3 channels)
[11]	LSB +Chaotic (RGB cover)	44.605	9 BPP (3 bits for each channel)
[12]	LSB +PVD + Chaotic (GrayScale)	37.38	2,365 BPP
[13]	LSB + PVD (RGB Cover)	40.4	3.1 BPP
Proposed	LSB+ Chaotic +Fractal (Gray Scale)	45.6	2,93 BPP (for Single Channel)

proposed system has a theoretical computational complexity of 2^n which is also the property of NP-Hard problems; keyspace is as large as 2^{359} which is considered to be secure enough according to [26].

There are some attacking scenarios that will be discussed briefly namely; Known Cover Attack, Known Message Attack, Known Stego Attack and Stego-Only Attack.

Known Cover Attack: This is the case when the attacker knows the original cover image. In this scenario it is a trivial job to compare stego and original cover and in this worst-case attacker may reveal the very existence of the secret message using statistical analysis. However, with the proposed technique it will be infeasible to recover the original secret message.

Known Message Attack: in this scenario, the attacker knows the secret message. Even it may help to analyze, it will be still difficult since the message is inserted at arbitrary locations.

Known Stego Attack: in this scenario steganography the attacker knows the algorithm. This is a well-accepted and known principle of Auguste Kerchoff. Even attacker has prior knowledge about the proposed system s/he still needs to search for the key in an astronomic keyspace of 2^{359} .

Stego-Only Attack: in this scenario, only the stego image is intercepted in the communication of the parties. It is still far difficult to restore the original message.

Jessica Fridrich [39] states that the length of the message in images is very related to the detection of the secret message. In other words, a small fragment of secret message inserted in a relatively large carrier will result in a tiny percentage manipulation and it will be harder to detect such artifacts.

3) COMPARISON WITH OTHER STUDIES

In order to compare the achieved results better, similar techniques having similar results are gathered in Table 6. As it can be observed clearly that the proposed system performs better than compared studies regarding image quality metric PSNR and capacity. Only those common metrics are compared amongst the studies.

V. CONCLUSION

Using chaotic Logistic map and Mandelbrot fractal sets in conjunction with Huffman encoding–compression, in this paper, a new secure, efficacious and competent high payload capacity steganographic algorithm enhanced with a new encryption algorithm and morphology-based adaptive region selection is presented. The results indicate that the proposed algorithm surpasses regarding imperceptibility (better quality index and PSNR values), data hiding capacity, besides it offers security. The algorithm is stressed with visual and the chi-square test to assess the resistance of such attacks and it is concluded that the algorithm is exempt such attacks. Since only the last bit of the pixel is used to insert secret information, after it is hidden the quality of the stego image is still very good.

At the end of study some guidelines can be listed as contribution like Morphology of the LSB plane of cover image must be taken into account before embedding, If encryption scheme is going to be presented it should diffuse the secret message but uniformity should not be looked for since the main steganalysis tools look for uniformity in the PoV, Maximum embedding capacity should not be reached since the lesser message embedded lesser probability of detection.

The proposed algorithm has experimented in grayscale images however it is not limited to grayscale and can be extended to color images with little algorithm modification since every color channel can be accepted as grayscale. For future work if it is implemented so an increase in the capacity of hiding is expected to be as high as three times.

ACKNOWLEDGMENT

For certain parts of the presented study patent application is pending with file number 2019/13963 from Turkish Patent and Trademark Office, which is a department of Republic of Turkey Ministry of Industry and Technology.

REFERENCES

- [1] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Hallorana, "Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research," *Neurocomputing*, vol. 335, pp. 299–326, Mar. 2019.
- [2] H. Wang and S. Wang, "Cyber warfare: Steganography vs. Steganalysis," *Commun. ACM*, vol. 47, no. 10, pp. 76–82, 2004.
- [3] B. Li, J. He, J. Huang, and Y. Q. Shi, "A survey on image steganography and steganalysis," *J. Inf. Hiding Multimedia Signal Process.*, vol. 2, no. 2, pp. 142–172, 2011.
- [4] S. Venkatraman, A. Abraham, and M. Paprzycki, "Significance of steganography on data security," in *Proc. Int. Conf. Inf. Technol. Coding Comput. (ITCC)*, Apr. 2004, pp. 347–351.
- [5] C. H. Yang, C. Y. Weng, S. J. Wang, and H. M. Sun, "Adaptive data hiding in edge areas of images with spatial LSB domain systems," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 488–497, Sep. 2008.
- [6] X. Li, B. Yang, D. Cheng, and T. Zeng, "A generalization of LSB matching," *IEEE Signal Process. Lett.*, vol. 16, no. 2, pp. 69–72, Feb. 2009.
- [7] M. Khodaei and K. Faez, "New adaptive steganographic method using least significant-bit substitution and pixel-value differencing," *IET Image Process.*, vol. 6, no. 6, pp. 677–686, Aug. 2012.
- [8] J. Wang, J. Ni, X. Zhang, and Y.-Q. Shi, "Rate and distortion optimization for reversible data hiding using multiple histogram shifting," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 315–326, Feb. 2017.
- [9] S. Das, S. Sharma, S. Bakshi, and I. Mukherjee, "A framework for pixel intensity modulation based image steganography," in *Proc. Adv. Comput. Intell. Eng.*, vol. 563, 2018, pp. 3–14. doi: 10.1007/978-981-10-6872-0_1.
- [10] E. Yavuz, R. Yazici, M. C. Kasapbaşı, and E. Yamaç, "A chaos-based image encryption algorithm with simple logical functions," *Comput. Elect. Eng.*, vol. 54, pp. 471–483, Aug. 2016. doi: 10.1016/j.compeleceng.2015.11.008.
- [11] Z. Man, J. Li, X. Di, and O. Bai, "An image segmentation encryption algorithm based on hybrid chaotic system," *IEEE Access*, vol. 7, pp. 103047–103058, 2019. doi: 10.1109/ACCESS.2019.2931732.
- [12] A. Anees, A. M. Siddiqui, J. Ahmed, and I. Hussain, "A technique for digital steganography using chaotic maps," *Nonlinear Dyn.*, vol. 75, no. 4, pp. 807–816, 2014.
- [13] M. Aziz, M. H. Tayarani-N, and M. Afsar, "A cycling chaos-based cryptic-free algorithm for image steganography," *Nonlinear Dyn.*, vol. 80, no. 3, pp. 1271–1290, 2015.
- [14] S. Prasad and A. K. Pal, "Logistic map-based image Steganography scheme using combined LSB and PVD for security enhancement," in *Emerging Technologies in Data Mining and Information Security*, vol. 814, P. Dutta, J. Mandal, A. Bhattacharya, and S. Dutta, Eds. Singapore: Springer, 2018.
- [15] G. Swain, "A steganographic method combining LSB substitution and PVD in a block," *Proc. Comput. Sci.*, vol. 85, no. 85, pp. 39–44, 2016.
- [16] S. Rajendran and M. Doraipandian, "Chaotic map based random image steganography using LSB technique," *Int. J. Netw. Secur.*, vol. 19, pp. 593–598, Jul. 2017.
- [17] R. Roy, A. Sarkar, and S. Changder, "Chaos based edge adaptive image steganography," *Proc. Technol.*, vol. 10, pp. 138–146, 2013.
- [18] *Turkish National Corpus*. Accessed: Sep. 17, 2019. [Online]. Available: <https://v3.tnc.org.tr>
- [19] *TS Corpus*. Accessed: Sep. 17, 2019. [Online]. Available: <https://tsocorpus.com>
- [20] *Ikipedia Letter Frequency*. Accessed: Sep. 17, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Letter_frequency
- [21] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. Inst. Radio Eng.*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952.
- [22] D. S. Stinson, *Cryptography: Theory and Practice*. Boca Raton, FL, USA: CRC Press, 1995, pp. 25–31.
- [23] B. Mandelbrot, *Fractals and Chaos: The Mandelbrot Set and Beyond*. New York, NY, USA: Springer, 2013.
- [24] Y. C. Kwun, M. Tanveer, W. Nazeer, K. Gdawiec, and S. M. Kang, "Mandelbrot and julia sets via Jungck-CR iteration with s -convexity," *IEEE Access*, vol. 7, pp. 12167–12176, 2019.
- [25] T. Sun and D. Wang, "The symmetry in the noise-perturbed mandelbrot set," *Symmetry*, vol. 11, no. 4, p. 577, Apr. 2019.
- [26] L. Chen, S. Jordan, Y. K. Liu, D. Moody, R. Peralta, R. Perlnar, and D. Smith-Tone. (2016). *Report on Post-Quantum Cryptography*. [Online]. Available: <https://nvlpubs.nist.gov/nistpubs/ir/2016/nist.ir.8105.pdf>
- [27] A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," in *Information Hiding (Lecture Notes in Computer Science)*, vol. 1768, A. Pfitzmann, Ed. Berlin, Germany: Springer, 1999.
- [28] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. London, U.K.: Pearson Education, 2008, pp. 627–688.
- [29] T. Saha, S. Sengupta, and T. Dasgupta, "Chaotic cipher based spatial domain steganography with strong resistance against statistical attacks," in *Proc. 3rd Int. Conf. Res. Comput. Intell. Commun. Netw. (ICR-CICN)*, Kolkata, India, Nov. 2017, pp. 365–370. doi: 10.1109/ICR-CICN.2017.8234536.
- [30] T. Yang, C. W. Wu, and L. O. Chua, "Cryptography based on chaotic systems," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 44, no. 5, pp. 469–472, May 1997.
- [31] J. K. Mandal and S. Das, "An information hiding scheme in wavelet domain using chaos dynamics," *J. Sci. Ind. Res.*, vol. 77, no. 5, pp. 264–267, May 2018.
- [32] M. C. Kasapbaşı and W. Elmasry, "New LSB-based colour image steganography method to enhance the efficiency in payload capacity, security and integrity check," *Sadhana*, vol. 43, no. 5, p. 68, May 2018.
- [33] R. A. Elmanfaloty and E. Abou-Bakr, "Random property enhancement of a 1D chaotic PRNG with finite precision implementation," *Chaos, Solitons Fractals*, vol. 118, pp. 134–144, Jan. 2019.

- [34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [35] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Process. Lett.*, vol. 9, no. 3, pp. 81–84, Mar. 2002.
- [36] C. Stanley, "Pairs of Values and the Chi-squared Attack," M. S. thesis, Dept. Math., Iowa State Univ., Des Moines, IA, USA, 2005.
- [37] *The USC-SIPI Image Database*. Accessed: Sep. 17, 2019. [Online]. Available: <http://sipi.usc.edu/database/>
- [38] *NASA Image and Video Library*. Accessed: Sep. 17, 2019. [Online]. Available: <https://images.nasa.gov>
- [39] J. Fridrich, M. Goljan, and D. Hoge, "Attacking the outguess," in *Proc. 3rd Inf. Hiding Workshop Multimedia Secur.*, Juan-les-Pins, France, 2002.



MUSTAFA CEM KASAPBAŞI was born in Trabzon, Turkey, in 1976. He received the B.S., M.S., and Ph.D. degrees in computer and control education from Marmara University, in 1999, 2001, and 2009, respectively. He has been a full-time Assistant Professor with Computer Engineering Department, Engineering Faculty, Istanbul Commerce University, since 2010. He has authored or coauthored many peer-reviewed journal articles and actively working as a reviewer for scholarly journals. His research interests include image encryption, steganography steganalysis, and data mining.

• • •