Computer Science and Information Systems 18(4):1311–1332 https://doi.org/10.2298/CSIS210227048T

Compensation of Degradation, Security, and Capacity of LSB Substitution Methods by a new Proposed Hybrid n-LSB Approach

Kemal Tütüncü and Özcan Çataltaş*

Faculty of Technology,Selcuk University,
42130 Konya, Turkey
{ktutuncu, ozcancataltas}@selcuk.edu.tr

Abstract. This study proposes a new hybrid n-LSB (Least Significant Bit) substitution-based image steganography method in the spatial plane. The previously proposed n-LSB substitution method by authors of this paper is combined with the Rivest-Shamir-Adleman (RSA), RC5, and Data Encryption Standard (DES) encryption algorithms to improve the security of the steganography, which is one of the requirements of steganography, and the Lempel-Ziv-Welch (LZW), Arithmetic and Deflate lossless compression algorithms to increase the secret message capacity. Also, embedding was done randomly using a logistic map-based chaos generator to increase the security more. The classical n-LSB substitution method and the proposed hybrid approaches based on the previously proposed n-LSB were implemented using different secret messages and cover images. When the results were examined, it has been seen that the proposed hybrid n-LSB approach showed improvement in all three criteria of steganography. The proposed hybrid approach that consists of previously proposed n-LSB, RSA, Deflate, and the logistic map had the best results regarding capacity, security, and imperceptibility.

Keywords: image steganography, lossless compression, logistic map, data encryption.

1. Introduction

Today, the use of the internet and other technological tools in communication between people is widespread. According to a survey conducted among OECD countries in 2019, internet access on an individual basis has increased from 45.7% to 85.6% between 2005-2018[1]. As the use of the internet increases in communication between people, privacy concerns also increase. For this reason, efforts to ensure privacy in communication have increased.

Steganography is one of the data hiding sciences that aims to increase confidentiality in communication [2-3]. The primary purpose of steganography is to conceal the existence of a secret message. This purpose is the most crucial feature that distinguishes it from other data hiding sciences. Since a media file containing a secret message will

* Corresponding author

not be attracted by the third party viewing the message, the secret message will not arise. Therefore, researchers' interest in this subject has increased gradually.

Unlike other data hiding methods, the message is hidden in another media file called a cover or carrier in steganography. This file type can be text, image, audio, video and, etc. The embedded message received by the recipient is converted to the original message by reverse conversion. Since the image is used more in communication between people, the studies have focused on the image file as cover media [4].

Cryptography and watermarking are two other concepts used to provide digital information security. In cryptography, the encrypted output of the encryption algorithm attracts the third person's attention to extract the original message[5]. On the other hand, although both techniques have a data hiding scheme, the intent of watermarking is different from steganography. Steganography aims to conceal the existence of any secret message, while watermarking makes it challenging to remove or manipulate the message.

Steganography algorithms can generally be divided into two categories: spatial domain and transform domain [6]. In the spatial domain, the secret message's bits are embedded into the cover image by directly manipulating the pixel values. On the other hand, in the transform domain, a secret message is placed in the frequency coefficients calculated from the cover image's pixel values using some mathematical functions. The methods applied in the spatial domain have less computation and time complexity but are relatively less resistant to some attacks. The algorithms in the spatial domain have a very high embedding capacity with very poor perceptibility.

In practice, while designing a steganography algorithm, three main features must be considered carefully: imperceptibility, embedding capacity, and security [7]. The embedding capacity and imperceptibility of the stego image are inversely proportional. As the embedding capacity increases, the quality of the stego image decreases. Therefore, using compression methods before embedding the secret message will increase the capacity of the cover media and reduce the detectability of the secret message.

The third feature, security, provides resistance against attacks that is subject to steganalysis. Although steganography's main feature is that it is not suspicious, the message can be obtained in case of possible detection of embedding algorithm. Therefore, encrypting the secret message with known cryptology algorithms before embedding it will increase communication security.

Another way to improve security in steganography is to embed the secret message randomly instead of sequentially. For this purpose, the embedding process can be done with the help of numbers generated by random number generators [8]. In literature, pseudo-random generators and chaos-based generators are generally used as random number generators.

In this study, we have hybridized the different compression methods to increase the embedding capacity of the n-LSB substitution method we introduced in another study [9] and with different encryption methods to increase security. Additionally, we increased security by using a chaos-based (logistic map) embedding algorithm regarding compressed and encrypted messages. These hybrid approachesare tested in different size messages and different images, and the results were compared. It has been seen that the proposed hybrid system compensated degradation, security, and capacity of classical n-LSB based image steganography.

The paper is organized as follows: In the second part, the existing studies in the literature are examined. In the third part, classical n-LSB substitution method, data compression methods, data encryption methods, random number generators, and image quality evaluation methods are mentioned. In the fourth section, the n-LSB substitution method [9] and the proposed hybrid methods are explained. The obtained results are shown in the fifth chapter. In the sixth section, the results are interpreted, and suggestions are made about future works.

2. Related Works

In this study, the proposed hybrid methods have been compared with the classical LSB substitution method as can be seen in the following section. Thus, we will include studies in the literature that modified the LSB substitution method or combined it with compression and encryption methods.

In our previous study [9], the classical n-LSB substitution method was improved and a new version of n-LSB was proposed and tested on different images. Obtained stego images were compared with stego images obtained by the classical n-LSB substitution method. The proposed n-LSB method caused an increase of 6.6% in the Peak Signal to Noise Ratio (PSNR) value regarding the classical n-LSB substitution method.

In their study, Rajput et al. used RSA cryptography and Spatial Orientation Tree Wavelet (STW) compression methods for hiding a secret message in color and gray-scale images. The secret message was encrypted using the RSA encryption algorithm, then embedded in the cover image compressed by the STW compression algorithm. They tested their method using 8 different cover images and obtained PSNR values ranging from 77.3 dB to 83.9dB [10].

Chen has proposed a new module-based LSB substitution method. In this method, the repeated bits in the secret message are detected and the repeated bits are coded with a code. He tested his method by hiding 7 different gray-scale images at 256x512 pixel resolution in 2 different gray-scale images at 512x512 pixel resolution and obtained the PSNR values ranging from 34dB to 36dB in the test result [11].

Akhtar and colleagues [12] proposed a new module-based LSB steganography method by developing the algorithm proposed by Chen [11]. They tested their method by hiding 10 different gray-scale images with 256x256 pixel resolution in 2 different gray-scale cover images with 512x512 pixel resolution. They obtained PSNR values ranging from 34dB to 40dB in the test result. According to the classical LSB method, they obtained increases of between 3% and 25% regarding PSNR. At the same time, they also applied the method suggested by Chen and emphasized that they achieved a higher PSNR value in their method.

Chikouche combined the classical LSB substitution method with the Advanced Encryption Standard (AES) cryptography method and the Deflate compression method in his work. The LSB substitutionmethod was implemented randomly with a pseudo-random generator rather than sequentially. They embedded 3264-bit data in a color cover image with 512x512 pixel resolution and emphasized that their method is better than according to the security criterion [13].

In their study, Manjula and Shivakumar compressed the message they encrypted with AES and Elliptic Curve Cryptography (ECC) with the LZW algorithm and embedded it with the classical LSB substitution method. 32-bit messages were hidden in different images and the PSNR values ranging from 79dB to 81dB values were obtained. Then the messages with a length ranging from 32 bits to 288 bits were hidden in different images and the PSNR values ranging from 77dB to 81dB were obtained. Also, they stated that they have 2 times security because the message is encrypted twice [14].

Kasapbaşi proposed a new image steganography scheme including chaos-based Huffman encoding algorithm and fractal encryption. Firstly, he calculated the frequency of the alphabets and other characters in a section of Turkish newspaper and encoded them with Huffman encoding. He encoded the compressed text with random numbers generated by the logistic map. The message was embedded in the selected LSBs of the cover image. The proposed method was found to be successful in terms of encryption [15].

Rachmawanto et al. proposed a hybrid method consists of the AES cryptology method and classical LSB substitution method. They tested their method and obtained PSNR values ranging from 58 dB to 80dB [16].

SupriadiRustad et al. proposed a new image steganography method based on finding an adaptive pattern in inverted LSB steganography. They obtained thePSNR value ranges from 52.49 to 57.45, and the SSIM ranges from 0.9991 to 0.9999 [7].

3. Materials and Methods

3.1. LSB Substitution Method

The basic principle of the LSB substitution method is to replace the LSB of each pixel with the message bit in the order of the cover image[17]. It can be applied to RGB or gray-scale images. The value of each pixel, which consists of 8 bits, 0 to 255, is either increased by 1, decreased by 1, or unchanged. A change of ±1 in the image pixel will not make a big difference on the image.

3.2. Data Compression Methods

According to the compression formats, data compression methods are divided into two categories: lossy compression and lossless compression [18]. If the original data can be recovered without any changes after compressing the data, this type of compression is called lossless compression. Lossless compression methods ensure that the original data is preserved precisely and that no detail is desired to be lost. In the other category of compression algorithms, lossy compression, original data cannot be obtained precisely after the recovery. In this study, LZW, Arithmetic, and Deflate algorithms had been chosen to compress the message before it was hidden. Detailed information about these algorithms is shown below.

Lempel-Ziv-Welch

The LZW algorithm is a compression method derived from the LZ78 algorithm [18]. It was discovered in 1984 by Terry A. Welch and introduced in his paper titled “A Technique for High-Performance Data Compression” (1984).

There is no preset dictionary in the LZW compression method. Dictionary is created dynamically according to the context to be compressed. For this reason, when the LZW method is used to compress the secret message in steganography, the sender does not need to transmit a dictionary to the recipient. Once the recipient has extracted the compressed message from the stego image, the dictionary will be dynamically created, and the secret message will be obtained.

Arithmetic

The primary purpose of arithmetic coding is to assign an interval to each character. Then, this range is assigned a decimal number. The algorithm starts with 0 and 1 intervals. After reading each character in the input data, the interval is divided into subparts as a smaller range than the input character's probability. This sub-range becomes the new range and is partitioned according to the probability of that character. This process is repeated for each input character. When this is done, every floating point in the last interval uniquely represents the input data [18-20].

Deflate

Deflate is a popular compression method used in well-known algorithms such as Zip and Gzip. Deflate method is used by many important programs such as PNG image, HTTP protocol, and PDF. The Deflate method is a dictionary-based compression technique based on LZ77 and Huffman coding. There are three different modes in Deflate method. In the first mode, input symbols are subdivided without compression. This mode is used for non-compressible files or when someone wants to partition a file without compression. The second mode is a single-pass compression solution. In this mode, a predetermined coding table is used during coding. This mode is used in real-time applications [21]. The third mode of Deflate is a two-pass compression solution based on the dictionaries produced according to the statistical properties of the input file.

3.3. Data encryption methods

RSA

The RSA encryption algorithm was proposed by Ron Rivest, Adi Shamir, and Leonard Adleman in 1977 [22]. The expansion of the RSA consists of the initials of the names of the developers. The RSA algorithm is one of the asymmetric encryption methods. Two different keys are used for encryption and decryption, but these two keys are related to each other. The key used for encryption is a public key and is known by everyone. The key used for decryption is a private key and is only found on the receiving side [23].

The encryption steps of the RSA algorithm are as follows:

1. Two prime numbers, such as p and q, are input parameters.
2. The value of $n = p * q$ is the base value, and $\varphi(n) = (p-1) * (q-1)$Euler valueis calculated.
3. The number of e (public keys) is selected as $1 < e < \varphi(n)(\varphi(n)$isa prime number).
4. d value is selected so that $d * e = 1\ mod(\varphi(n))$. This value is a private key.
5. The $c = m^e\ mod(n)$ formula encrypts each message character.

To extract an encrypted message using the RSA algorithm, the first four steps are applied in the same way, andthen the secret message is obtained with the formula $m = c^d\ mod(n)$.

RC5

The RC5 algorithm is one of the symmetric encryption algorithms. It was introduced by Ron Rivest in 1994. The RC5 algorithm is simple to implement because it uses basic mathematical and logical operators. Furthermore, the variable key length distinguishes RC5 from traditional encryption methods such as Data Encryption Standard (DES). The implementation steps of the RC5 encryption algorithm are presented below [24, 25]:

1. Firstly, define *w*, *r*, and Key parameters.
2. Obtain *P* and *Q* constants.

```
  P=odd(e-2)2w
  Q=odd(Φ-2)2^w
```

3. Convert Key *K* byte to words.

```
for i=b-1 to 0
    L[i/u] = (L[u/i] << 8) + K[i]
```

4. Initialize key-independent array, *S*.

```
S[0] = P
for i = 1 to 2(r+1)-1
    S[i] = S[i-1] + Q)
```

5. Mix secret key in the L and S array.

```
i = j = 0
A = B = 0
do 3 * max(t, c) times:
    A = S[i] = (S[i] + A + B) << 3
    B = L[j] = (L[j] + A + B) << (A + B)
i = (i + 1) % t
    j = (j + 1) % c
```

6. Divide the input text into w-bit blocks (A and B are two of these blocks) and encrypt each block.

```
A = A + S[0]
B = B + S[1]
for i = 1 to r do:
    A = ((A ^ B) << B) + S[2 * i]
    B = ((B ^ A) << A) + S[2 * i + 1]
```

7. Decrypt using A and B.

```
for i = r down to 1 do:
    B = ((B - S[2 * i + 1]) >> A) ^ A
    A = ((A - S[2 * i]) >> B) ^ B
B = B - S[1]
A = A - S[0]
```

Data encryption standard (DES)

DES was one of the symmetric encryption methods introduced by the National Institute of Standards & Technology (NIST) in 1976 for all government communications. It has also been used for a long time in bank transactions[26].

In the DES algorithm, the input text is divided into blocks. Each has a 64-bit message. A 64-bit key is required in the encryption process.Eight bits of this key are used as parity bits. Encryption is done in 16 rounds.Each round uses a new key that is the 48-bit length. These keys are obtained using the input key. The block diagram of the DES algorithm is shown in Fig.1.

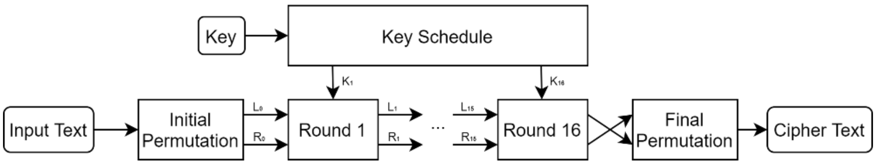

Fig. 1. Block diagram of DES encryption algorithm

3.4. Chaos generator

A random number is a series of numbers or symbols that are not predictable with random luck and do not repeat in a particular pattern. Random number generators have many field-critical presets, such as secure communications, data transmission, and storage. Random number generators are examined in two categories: pseudo-random generators and real random generators.

Chaos-based generators are used more extensively than pseudo-random generators because they are real random generators. Since the chaos generators are very sensitive to input parameters, the numbers they will produce constantly are not predictable. For this reason, it is frequently used in information security applications [27].

One of the simple chaos systems widely used and applied in the literature is the logistic map. The formula of the logistic map is as follow:

$$x_{k+1} = \mu * x_k * (1 - x_k) \tag{1}$$

Here, x_0 and μ are input parameters. When $3.57 < \mu \leq 4$, the system goes into a chaotic state and generates random numbers.

3.5. Image Quality Evaluation Criteria

Image evaluation criteria are methods used to learn the amount of change in the cover image. The image evaluation criteria used in this study are explained below.

Peak Signal-to-Noise Ratio (PSNR)

PSNR is one of the essential criteria used to evaluate image quality. The PSNR is the ratio of the power of the highest possible power of the cover image to the power of the difference between the cover image and the stego image. A high PSNR value means little distortion in the stego image, while a low PSNR value means more distortion in the stego image [28].

The PSNR value can be calculated using the following formula:

$$PSNR = 20 * log\frac{255}{\sqrt{MSE}} \tag{2}$$

$$MSE = \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{\left(S(i,j) - I(i,j)\right)^2}{m * n} \tag{3}$$

Average Difference

The average difference (AD) metric equals the mean of the sum of the differences between the cover image pixels and the stego image pixels. The low average difference value means that there is less distortion in the stego image.

The average difference formula is:

$$AD = \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{S(i,j) - I(i,j)}{m * n} \tag{4}$$

Universal Image Quality Index (UIQI)

UIQI (Universal Image Quality Index) is an index that attempts to model any distortion on the image[29]. These distortions can be in the form of a combination of the following three factors: correlation, luminance distortion, and contrast distortion. The UIQI value is between [-1, 1]. 1 means the images are identical.UIQI formula is shown in (5).

$$UIQI = \frac{\sigma_{xy}}{\sigma_x\sigma_y} * \frac{2\bar{x}\bar{y}}{\bar{x}^2 + \bar{y}^2} * \frac{2\sigma_x\sigma_y}{{\sigma_x}^2 + {\sigma_y}^2} \tag{5}$$

Here, σ denotes standard deviation and $\bar{x}$and $\bar{y}$ denote the average of the cover and stego images, respectively.

4. Proposed Methods

The classical 1-LSB, 2-LSB, and 3-LSB substitution methods are the most common methods used in steganography because of their ease of implementation and high capacity. Many different methods developed are built on these methods. However, they need to be improved because it is easy to detect, and third-person can directly access the message when it is detected.

4.1. Proposed n-LSB Method

The pseudo-code of the application steps of the n-LSB substitution method [9] by the author of this paper is presented in Fig. 2. The proposed n-LSB method reduces the excess change between the cover and stego image pixels while applying the classical n-LSB substitution methods. The proposed n-LSB method has only been applied to 2-LSB and 3-LSB methods because there is no effect on the 1-LSB method and also 4 and more bit substitution methods that are not used in the literature. The 2 and 3-bit implementation of the proposed n-LSB method is described and illustrated below.

Proposed 2-LSB substitution method

When $n = 2$ in the pseudo-code given in Fig. 2, the secret message is embedded in the cover image using the classical 2-LSB substitution method. At the end of the embedding process, the difference between the cover image and the stego image pixels is examined. Since at most two LSBs of the cover image pixels can be changed, the decimal difference will be one of 3, 2, 1, 0, -1, -2, -3. If this difference is -2, -1, 0, 1, or 2, then the pixel of the stego image is left unchanged. In another case, if this difference is 3, all bits from the 3rd LSB of the stego image are examined, the first 0 is converted to 1, and the previous 1s are converted to 0. So the related pixel of the stego image is added to the decimal number 4, the difference between the pixels falls from 3 to -1. If there are no 0's between the 3rd and 8th bits (most significant bit), the related pixels are left unchanged in the stego image. If the difference is -3, all bits from the 3rd bit of the related pixel are examined, the first 1 encountered is converted to 0, and the previous 0s are converted to 1. Thus, the decimal number 4 is subtracted from the corresponding pixel, and the difference between the pixels falls from -3 to 1. If there is no 1 between the 3rd and 8th bits, then the corresponding pixel is left unchanged in the stego image. With this method, the deterioration of the pixels where the degradation is excessive is reduced [9]. Examples of the implementation of the proposed 2-LSB are shown in Table 1.

```
Input:C,S,n
Output: S
C: Cover image pixels
S: Stego image pixels
n: Embedding method (n=2, 3)
$C(p)_i$,$S(p)_i$: ith bit of related pixel; i=1,..,8
For each p inevery pixel of C,S
If$C(p) - S(p) > 2^{(n-1)}$
        If$S(p)_{n+1} = 0$
        $S(p)_{n+1} = 1$
Elseif$S(p)_{n+2} = 0$
        $S(p)_{n+2} = 1$
                $S(p)_{n+1} = 0$
…
Elseif$S(p)_8 = 0$
                $S(p)_8 = 1$
          $S(p)_7 = \cdots = S(p)_{n+1} = 0$
End If
Elseif$C(p) - S(p) < -2^{(n-1)}$
        If$S(p)_{n+1} = 1$
        $S(p)_{n+1} = 0$
        Elseif$S(p)_{n+2} = 1$
                $S(p)_{n+2} = 0$
                $S(p)_{n+1} = 1$
…
Elseif$S(p)_8 = 1$
                $S(p)_8 = 0$
          $S(p)_7 = \cdots = S(p)_{n+1} = 1$
End If
End If
End For each
```

Fig. 2. The pseudo-code of the proposed n-LSB substitution method

In Table 1, randomly generated 2-bit messages were hidden in the randomly generated 20 pixels cover image pixels consisting of 8 bits by the 2-LSB method. After the embedding process, the differences between the pixels of the cover image and the stego image were examined. If the difference is -3 or 3, the proposed method was applied. As shown in the table, the difference in 5 of 20 pixels is -3 or 3, so the pixels outside these 5 pixels were not changed. The AD before the enhancement was 1.2, but after the enhancement, this difference was reduced to 0.9. Thereby the amount of distortion in the image was reduced.

Table 1. Proposed 2-LSB substitution example. The changed bits of cover image pixel obtained after embedding with the 2-LSB method and the bits of stego image pixel obtained after applying the proposed compensation method were shown in red and green color, respectively

Pixel No	Cover image	Message to be embedded (2-bit)	Stego image	Difference	New stego image	New difference
1	01111001	01	011110***01***	0	01111001	0
2	00100011	10	001000***10***	1	00100010	1
3	10010000	11	100100***11***	-3	100***01***11	1
4	00000001	10	000000***10***	-1	00000010	-1
5	11000011	00	110000***00***	3	11000***1***00	-1
6	11000010	11	110000***11***	-1	11000011	-1
7	00101110	01	001011***01***	1	00101101	1
8	11111000	01	111110***01***	-1	11111001	-1
9	00110000	10	001100***10***	-2	00110010	-2
10	11010100	11	110101***11***	-3	11010***0***11	1
11	11100000	11	111000***11***	-3	11***0111***11	1
12	00100000	10	001000***10***	-2	00100010	-2
13	01010001	01	010100***01***	0	01010001	0
14	01101101	10	011011***10***	-1	01101110	-1
15	11101001	10	111010***10***	-1	11101010	-1
16	11001101	01	110011***01***	0	11001101	0
17	01110111	00	011101***00***	3	0111***10***00	-1
18	01111110	10	011111***10***	0	01111110	0
19	01001101	10	010011***10***	-1	01001110	-1
20	10000011	10	100000***10***	1	10000010	1

Proposed 3-LSB substitution method

When $n = 3$ in the pseudo-code given in Fig. 2, the secret message is embedded in the cover image using the classical 3-LSB substitution method. At the end of the embedding process, the difference between the pixels of the cover image and the stego image is examined. Since at most three LSBs of the cover image can be changed, the difference will get one of the values from -7 to 7. If the decimal difference between the pixels is -4, -3, -2, -1, 0, 1, 2, 3, or 4, then the pixel of the stego image is left unchanged. If the difference is 5, 6, or 7, all bits from the 4th bit of that pixel are examined, the first 0 value encountered is converted to 1, and the previous 1s are converted to 0. So the related pixel is added to the decimal number 8, and the difference between the pixels falls from 5, 6, 7 to -3, -2, and -1, respectively. If there is no 0 value between the 4th and 8th bits, the related pixel of the stego image is left unchanged. Similarly, if the difference is -5, -6, or -7, all bits of the pixel are examined from the 4th bit, the first 1 value encountered is converted to 0, and the previous 0s are converted to 1. Thus, the decimal number 8 is subtracted from the related pixel, the difference between the pixels falls from -5, -6, -7 to 3, 2, and 1, respectively. If there is no 1 value between the 4th and 8th bits, then the related pixel is left unchanged. With this method, the deterioration of the pixels where the degradation is excessive is reduced [9]. Examples of the implementation of the proposed compensation method for 3-LSB are presented in Table 2.

Table 2. Proposed 3-LSB substitution example. The changed bits of cover image pixel obtained after embedding with the 3-LSB method and the bits of stego image pixel obtained after applying the proposed compensation method were shown in red and green color, respectively

Pixel No	Cover image	Message to be embedded (3-bit)	Stego image	Difference	New stego image	New difference
1	01111001	111	01111***111***	-6	0111***0***111	2
2	00100011	001	00100***001***	2	00100001	2
3	10010000	111	10010***111***	-7	100***01***111	1
4	00011111	010	00011***010***	5	00***100***010	-3
5	11000011	100	11000***100***	-1	11000100	-1
6	11000010	111	11000***111***	-5	1***0111***111	3
7	00101110	111	00101***111***	-1	00101111	-1
8	11111000	001	11111***001***	-1	11111001	-1
9	00110000	010	00110***010***	-2	00110010	-2
10	11010100	110	11010***110***	-2	11010110	-2
11	11100111	000	11100***000***	7	1110***1***000	-1
12	00100000	101	00100***101***	-5	00***011***101	3
13	01010001	101	01010***101***	-4	01010101	-4
14	01101101	101	01101***101***	0	01101101	0
15	11101001	110	11101***110***	-5	1110***0***110	3
16	11001101	100	11001***100***	1	11001100	1
17	01110111	001	01110***001***	6	0111***1***001	-2
18	01111110	111	01111***111***	-1	01111111	-1
19	01001111	001	01001***001***	6	010***10***001	-2
20	10000011	001	10000***001***	2	10000001	2

In Table 2, randomly generated 3-bit messages were hidden on the randomly generated 20 cover image pixels consisting of 8 bits by the 3-LSB method. After the embedding process, the differences between the pixels of the cover image and the stego image were examined. If the difference is -7, -6, -5, 5, 6, or 7, the proposed method was applied. As shown in the table, the difference in 9 of 20 pixels is -7, -6, -5, 5, 6, or 7, so the pixels outside these 9 pixels were not changed in the stego image. The AD before the enhancement was 3.45, but after the enhancement, this difference was reduced to 1.85. Thereby the amount of distortion in the stego image was reduced.

4.2. Proposed hybrid-1

With the proposed n-LSB method [9], the degradation of pixels of the stego image has been reduced, which is one of the three basic principles of steganography. However, there has been no change in the other principles of steganography, which are security and capacity. To improve these two criteria, existing compression and encryption methods are combined with the proposed n-LSB method.

Compressing the secret message before embedding it in the cover image will reduce the degradation of the stego image as it will reduce the number of secret message bits to be embedded and increase the message length that can be embedded on the cover image. Also, since the 3rd person will not know the used compression algorithm, the secret message will not be directly available, even if the hidden data in the stego image is recovered. For this purpose, the message is compressed using text compression algorithms before being embedded. Three different algorithms, LZW, Arithmetic, and Deflate, have been applied as text compression algorithms and compared among themselves.

Embedding the message sequentially in the cover image makes it easier to extract it by the 3rd person. Encrypting the message before embedding it is a big solution, but since the embedding process is sequential, the 3rd person can quickly get the encrypted message. Although cryptography methods are challenging to break, it is not impossible to break. For this reason, the secret message may be passed on to other people. To overcome this problem, the message in steganography is often randomly embedded in the image with various random number generators rather than sequentially. In our proposed hybrid n-LSB approach, random numbers are generated with the logistic map-based chaos generator to overcome this problem, and the message is randomly embedded in the cover image. The x_0 and μ values of the chaotic generator are used as input parameters. For this reason, these parameters must be transmitted to the recipient to extract the hidden message.

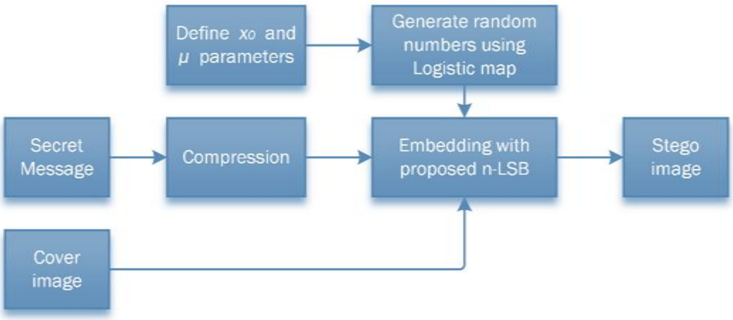
Define x_0 and μ parameters
Generate random numbers using Logistic map
Secret Message
Compression
Embedding with proposed n-LSB
Stego image
Cover image

Fig. 3. The flowchart of the proposed hybrid-1 method

To eliminate the shortcomings of the n-LSB method and obtain a safer embedding algorithm, twodifferent hybrid methods were created.In the first hybrid method, named as proposed hybrid-1, the secret message was compressed using Deflate compression algorithm, and then this compressed secret message was embedded to the cover image randomly using the proposed 2-LSB and 3-LSB substitution method and logistic map-based chaos generator. The reason for choosing Deflate compression algorithm is being superior to the other two methods in compression ratios according to applications presented in Results Section. The flowchart of the proposed hybrid-1 method was shown in Fig. 3.

4.3. Proposed hybrid-2

Encrypting the message before embedding it in the cover image will make it difficult to reach the secret message, even if the third party can completely get the information embedded in the image. For this purpose, the message is encrypted with different encryption algorithms. RSA, DES, and RC5 algorithms were tested and compared for this purpose.

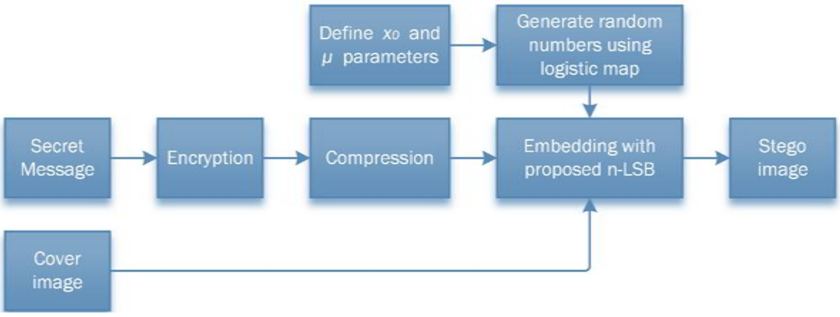
Define x_0 and μ parameters
Generate random numbers using logistic map
Secret Message
Encryption
Compression
Embedding with proposed n-LSB
Stego image
Cover image

Fig. 4. The flowchart of the proposed hybrid-2 method

In this hybrid method, named as proposed hybrid-2, in addition to the proposed hybrid-1 method, the secret message was encrypted before compressing process using the RSA encryption algorithm. The reasons why the RSA algorithm is chosen are its widespread use and being asymmetric encryption algorithm. Additionally, the RSA encryption algorithm is superior to the other two methods both in the average encryption time and in the file size to be encrypted per second according to applications presented in Results Section. The (p, q) values required to generate the public and private keys in RSA encryption are used as input parameters. For this reason, for the receiver to reach the secret message, the values p and q must be transmitted to the receiver. The flowchart of the proposed hybrid-2 method was shown in Fig. 4.

5. Results

In this section, the results obtained by applying the classical LSB substitution and proposed hybrid n-LSB approaches are evaluated. For this aim, three different text files and four different cover images were used. The cover images used are shown in Fig. 5. These images are RGB 24-bit images with a resolution of “Lena” 225x225, “Mandrill” 512x512, “Cat” 960x603, and “Peppers” 600x600 pixels. These images are in the ".bmp" file format. The methods we propose in this study are independent of these images and can be applied to any desired image without any constraints.The main reasons forchoosing these images as cover are being in different resolutions and well-known in the literature.

Three text files with sizes 6.95 kB, 13.59 kB,and 17.13 kB were selected as secret messages. These text files contain the standard English alphabet as well as some special characters (., ?, -, !, “”, ,).

When the results are evaluated, a comparison is made only with the classical LSB substitution method, as seen in the literature. The reason is that each of the methods in the literature is tested on different cover images using different messages. There is no common ground between methods proposed by other authors in the literature.

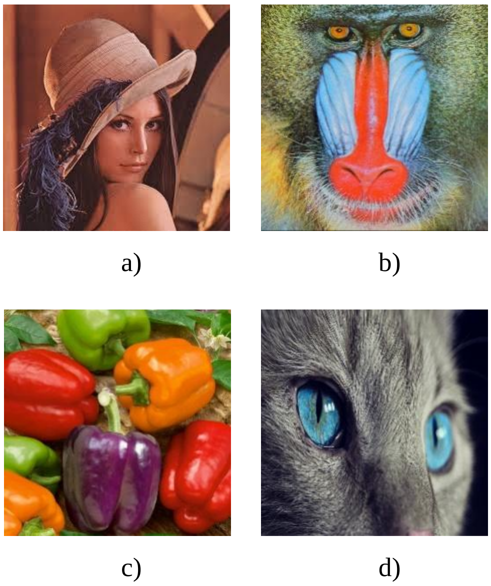

Fig. 5. Cover images used to test proposed methods

5.1. Comparison of data compression methods

In the proposed hybrid methods, the secret message file is compressed using different compression methods before being hidden in the cover image. The new message length and the compression rates resulting from the compression are shown in Table 3. The compression ratio values were calculated according to (6).

$$\%CompressionRatio = \frac{InputSize}{OutputSize} * 100 \tag{6}$$

The original message files with 56980, 111405, and 140364-bit lengths were compressed with the LZW algorithm; their size decreased to 38448, 72969, and 84643-bit, respectively. Compression ratios were obtained as 148.2%, 152.67% and 165.83% respectively. It is seen that as the message length increases, the compression ratio increases, and the highest compression ratio is obtained when the Text-3 file is compressed with LZW. The reason is that as the message length increases, the possibility of finding new words added to the dictionary increases. In other words, when there are 2, 3, or more character words added to the dictionary, the probability of encountering these words increases as the size of the message increases so that the coded words in the dictionary can be used instead of these words.

With the arithmetic algorithm, the secret message files, which are 56980, 111405, and 140364-bit length, have been reduced to the size of 35448, 69361, and 88109 bits respectively. Compression ratios were calculated as 160.74%, 160.62% and 159.31%, respectively. It is seen that as the message size increases, the compression ratio decreases, and the highest compression ratio is obtained when the Text-1 file is compressed with the arithmetic algorithm.

Table 3. Comparison of compression algorithms

Filename	Uncompressed size (bit)	LZW	Arithmetic	Deflate
		Compression ratio		
Text 1	56980	38448 148.20%	35448 160.74%	29496 193.18%
Text 2	111405	72969 152.67%	69361 160.62%	56832 196.03%
Text 3	140364	84643 165.83%	88109 159.31%	65744 213.50%

With the Deflate algorithm, the original message files, which are 56980, 111405, and 140364-bits in length, have been reduced to the size of 29496, 56832, and 65744 bits, respectively. Compression ratios were calculated as 193.18%, 196.03% and 213.50%, respectively. Since the Deflate algorithm is a hybrid algorithm consisting of Huffman and LZ77 codes, the highest compression ratios according to other algorithms were obtained with this algorithm. Also, as the message length increased, the compression ratio increased, and the highest compression ratio was obtained when the Text-3 file was compressed.

Since the highest compression ratio between LZW, Arithmetic, and Deflate algorithms is obtained by Deflate method, it is used as a compression method in the proposed hybrid-1 and hybrid-2 algorithms.

5.2. Comparison of data encryption methods

Determining the security and success of encryption algorithms is a complex process. To compare such algorithms on a common basis, encryption times are generally used. Therefore, the encryption times of the RSA, RC5, and DES encryption algorithms used in this study were calculated using texts of different lengths and are shown in Table 4.

Table 4. Comparison of encryption times of RSA, RC5, and DES

Text Size	RSA	RC5	DES
Text-1 (7122 byte)	10.23	24.91	21
Text-2 (13925 byte)	20.05	48.29	40.05
Text-3 (17545 byte)	24.95	60.97	50.42
Average	18.41	44.72	37.15
Bytes/sec	698.75	287.65	346.27

When Table 4 is examined, it can be seen that the RSA encryption algorithm is superior to other methods both in the average encryption time and in the file size to be encrypted per second.Therefore, RSA was chosen as the encryption method in the proposed hybrid-2 algorithm.

5.3. Test results

Three different secret message files were embedded in 4 different cover images using the classical LSB substitution method, the proposed n-LSB method [9], and proposed hybrid methods. The stego images were compared with the cover images by using image comparison criteria, and the results are shown in the sub-sections.

The following algorithms are used for embedding:

- Classical 1-LSB, 2-LSB, and 3-LSB methods
- Proposed n-LSB method (consist of proposed 2-LSB or 3-LSB methods) [9]
- Proposed hybrid-1 (consist of proposed 2-LSB or 3-LSB method combined with Deflate compression algorithm and logistic map-based random embedding method)
- Proposed hybrid-2 (consist of proposed 2-LSB or 3-LSB method combined with Deflate compression algorithm, RSA encryption algorithm, and logistic map-based random embedding method)

The input parameters used during embedding are:

- RSA encryption: $p = 3, q = 41$.
- Logistic map: $x_0 = 0.675$ and $\mu = 3.9763$.

PSNR

The cover and stego images were submitted to the PSNR test and the obtained results are shown in Appendix.Since the PSNR value is the ratio of the peak signal to the noise in the image, the higher PSNR value means that the image degradation is less. When the obtained results are examined, it is seen that the PSNR values of the proposed 2-LSB and 3-LSB methods are higher than the classical 2-LSB and 3-LSB methods. It can be said that the proposed hybrid-1 is better than the proposed n-LSB method and proposed hybrid-2. The second highest PSNR value was obtained by embedding Text-1 in Image-4 by the proposed hybrid-1 2-LSB method with 64.86235 dB and comes after the classical 1-LSB method. Also, the highest increase was 10.8426% (from 42.96577 dB to 47.62437 dB) when the Text-3 file was embedded to Image-1 by proposed hybrid-1 3-LSB compared to the classical 3-LSB method. It can be said that the highest PSNR increase can be achieved when the high-size message file is embedded into the low-resolution image.

Average Difference

The stego images and cover images obtained after applying the proposed and classical LSB substitution methods are compared according to the AD criterion, and the obtained results are presented in Appendix.The average difference is equal to the average of differences between the cover and stego image pixels. Since the stego image is desired to be similar to the cover image, it is expected that the average difference value is small. When the obtained results are examined, embedding Text-1 message on image-4 is obtained with the smallest mean difference value of 0.042 by the proposed hybrid-1 2-LSB algorithm. It is also seen that the proposed hybrid-1 is superior to other classical and proposed methods. However, as the resolution of the image increases or the length of the secret message decreases, the average difference value decreases.

UIQI

The stego images and cover images obtained after applying the proposed and classical LSB substitution method were compared according to the UIQI criterion, and the obtained results are shown in Appendix.It is preferable to have a high UIQI value because the difference between the stego image and the cover image is desired to be small. The classical 1-LSB method is the least corrupted method because it only changes the last pixels of the cover image. When we compare the proposed n-LSB method with the classical 2-LSB and 3-LSB algorithms, the proposed n-LSB method yields higher UIQI values. Additionally, the results obtained by the proposed hybrid-1 are compatible with the results obtained by the proposed n-LSB.However, since the UIQI value consists of a combination of correlation, luminance distortion, and contrast distortion, the results show differences in different cover images. Therefore, it is not possible to make a clear conclusion about which method is superior according to the UIQI value.

5.4. Capacity test

Capacity can be defined as the maximum amount of secret messages hidden in the cover image. Thus, it is essential to check the capacity of images when steganography methods are compared.

The effect of data compression on stego image capacity is shown in Table 5. The capacity values shown here are estimated values calculated from the compression ratios calculated in Table 3. Besides, since the application of encryption algorithms and other embedding methods such as proposed n-LSB and classical n-LSB do not affect the embedding capacity,only the classical 3-LSB method was used as an embedding method. According to Table 5, the highest capacity increase was achieved by the Deflate algorithm. With this algorithm, the capacity of image-4 was increased from 635.98 kB to 1357.81 kB, a 113.5% increase was obtained. Furthermore, according to the results shown in Table 3, it was obtained that the compression ratio increased as the message size increased. Accordingly, it is expected that with the LZW and the Deflate algorithm, the cover images will have a larger message capacity than the values shown in Table 5.

Table 5. Capacity test results

Image No	Uncompressed Capacity (kB)	Compressed Capacity (Expected) (kB)		
		LZW	Arithmetic	Deflate
Image 1	55.62	92.23	89.40	118.75
Image 2	288.00	477.59	462.93	614.88
Image 3	395.51	655.87	635.74	844.41
Image 4	635.98	1054.64	1022.27	1357.81

6. Conclusion and Discussion

In this paper, steganography methods which are one of the information security methods are examined, and a new hybrid method in image steganography is proposed. This method, which we propose, is based on the proposed n-LSB substitution method of the authors of this paper and tries to reduce the pixel differences between cover and stego image. In this regard, improvement in the perceptibility criterion, one of the three main criteria of steganography, has been achieved and confirmed with an implemented test.

Since the proposed n-LSB method is based on the LSB substitution method, the third party can extract the secret message easily. To solve this problem, instead of embedding the message bits in sequence, they are randomly embedded using a chaos-based random number generator. To increase the security a step ahead, RSA, RC5, and DES encryption algorithms are used to encrypt the secret message before being embedded. Then, data compression methods were combined with the proposed n-LSB method to provide improvement in both the capacity criterion and compensating for the increase in data size that would result in the use of encryption algorithms. Three compression methods, LZW, Arithmetic, and Deflate, were applied. The best compression ratio was obtained by the Deflate algorithm. For this reason, the secret message was compressed with the Deflate algorithm before being hidden in the cover image.

These proposed hybrid methods based on the proposed n-LSB method have been tested by hiding three message files in different sizes in 4 cover images with different resolutions and sizes. The highest PSNR value was obtained as 64.86 dB with proposed hybrid-1 (2-LSB), and the highest PSNR increase rate was 10.84% with proposed hybrid-1 (3-LSB) when the stego images and the cover images were compared according to image quality evaluation criteria. PSNR values were higher in all the different combinations of the proposed n-LSB method than in the classical LSB method. Moreover, the use of the Deflate compression algorithm in the proposed hybrid-1 method resulted in an increase of 113.5% in the embedding capacities of the cover images.

Thanks to the proposed hybrid methods, the shortcomings in using the n-LSB method have been eliminated, and more reliable methods have been obtained for data hiding. The proposed n-LSB and hybrid methods can be used regardless of the message and the cover image, as long as the secret message size does not exceed the capacity of the cover image.

The authors of this paper study the effects of the application of the combination of different compression and encryption algorithms with the proposed n-LSB method to different color spaces. Furthermore, the authors think that investigating the applicability of the proposed methods in the frequency domain will be a good research step.

Appendix

Table 6. Test results obtained by applying proposed methods

Metric	Image No	Text No	Classical			Proposed n-LSB		Proposed Hybrid-1		Proposed Hybrid-2	
			1-LSB	2-LSB	3-LSB	2-LSB	3-LSB	2-LSB	3-LSB	2-LSB	3-LSB
PSNR	Image 1	Text-1	55.37	51.30	46.85	53.62	49.69	54.74	50.42	52.79	48.57
		Text-2	52.46	48.41	43.98	50.69	46.82	52.21	48.06	50.52	46.49
		Text-3	51.46	47.39	42.97	49.68	45.83	51.72	47.62	50.04	46.10
	Image 2	Text-1	62.48	58.51	53.99	60.73	56.86	61.51	57.12	59.37	55.00
		Text-2	59.61	55.58	51.10	57.86	53.97	58.74	54.35	56.60	52.26
		Text-3	58.60	54.59	50.14	56.85	52.98	58.14	53.78	55.93	51.57
	Image 3	Text-1	63.83	59.82	55.36	62.06	58.10	62.87	58.44	60.74	56.24
		Text-2	60.93	56.91	52.44	59.09	55.19	60.04	55.60	57.83	53.45
		Text-3	59.96	55.89	51.48	58.03	54.13	59.46	55.01	57.14	52.73
	Image 4	Text-1	65.94	61.94	57.46	64.09	60.08	64.86	60.42	62.77	58.28
		Text-2	63.06	59.03	54.49	61.17	57.12	62.12	57.63	59.90	55.46
		Text-3	62.07	58.04	53.47	60.17	56.09	61.49	57.05	59.21	54.74
AD	Image 1	Text-1	0.0625	0.0793	0.1108	0.0627	0.0840	0.0443	0.0666	0.0703	0.1030
		Text-2	0.1226	0.1548	0.2155	0.1228	0.1634	0.0807	0.1167	0.1224	0.1708
		Text-3	0.1544	0.1953	0.2716	0.1548	0.2055	0.0907	0.1298	0.1379	0.1891
	Image 2	Text-1	0.0121	0.0152	0.0214	0.0121	0.0162	0.0092	0.0141	0.0151	0.0230
		Text-2	0.0236	0.0297	0.0417	0.0236	0.0316	0.0175	0.0267	0.0288	0.0435
		Text-3	0.0298	0.0375	0.0522	0.0298	0.0396	0.0200	0.0305	0.0336	0.0509
	Image 3	Text-1	0.0089	0.0112	0.0156	0.0089	0.0119	0.0067	0.0103	0.0110	0.0171
		Text-2	0.0174	0.0218	0.0305	0.0174	0.0232	0.0129	0.0199	0.0215	0.0327
		Text-3	0.0218	0.0275	0.0382	0.0220	0.0294	0.0148	0.0228	0.0252	0.0385
	Image 4	Text-1	0.0055	0.0069	0.0096	0.0055	0.0074	0.0042	0.0066	0.0069	0.0107
		Text-2	0.0107	0.0134	0.0190	0.0108	0.0146	0.0080	0.0125	0.0134	0.0207
		Text-3	0.0134	0.0169	0.0240	0.0136	0.0185	0.0092	0.0143	0.0157	0.0243
UIQI	Image 1	Text-1	0.9914	0.9789	0.9645	0.9859	0.9750	0.9908	0.9786	0.9863	0.9708
		Text-2	0.9867	0.9690	0.9441	0.9791	0.9612	0.9844	0.9683	0.9787	0.9595
		Text-3	0.9847	0.9629	0.9357	0.9750	0.9560	0.9829	0.9662	0.9771	0.9571
	Image 2	Text-1	1	0.9999	0.9999	1	1	0.9998	0.9997	0.9997	0.9995
		Text-2	0.9999	0.9995	0.9998	0.9999	0.9999	0.9996	0.9994	0.9994	0.9991
		Text-3	0.9999	0.9994	0.9997	0.9999	0.9999	0.9996	0.9993	0.9993	0.9989
	Image 3	Text-1	0.9994	0.9940	0.9985	0.9994	0.9991	0.9949	0.9938	0.9927	0.9904
		Text-2	0.9999	0.9991	0.9998	0.9999	0.9999	0.9996	0.9994	0.9993	0.9990
		Text-3	0.9971	0.9962	0.9945	0.9971	0.9962	0.9992	0.9885	0.9870	0.9825
	Image 4	Text-1	0.9989	0.9945	0.9989	0.9991	0.9991	0.9961	0.9933	0.9940	0.9869
		Text-2	0.9972	0.9922	0.9969	0.9976	0.9976	0.9937	0.9894	0.9909	0.9852
		Text-3	0.9961	0.9962	0.9958	0.9968	0.9968	0.9928	0.9885	0.9898	0.9840

References

1. OECD:ICT Access and Usage by Households and Individuals. [Online]. Available: https://stats.oecd.org/Index.aspx?DataSetCode=ICT_HH2 (current September 2021)
2. Luo, X., Wang, D., Wang, P., Liu, F.: A Review On Blind Detection For Image Steganography. Signal Processing, Vol. 88, No. 9, 2138-2157. (2008)
3. Chen, K., Lin, C., Zhong, S., Guo, L.:A Parallel SRM Feature Extraction Algorithm For Steganalysis Based On GPU Architecture. Computer Science and Information Systems, Vol. 12, No. 4, 1345-1359. (2015)

4. Karakus, S., Avci, E.:A New Image Steganography Method With Optimum Pixel Similarity For Data Hiding In Medical Images. Medical Hypotheses, Vol.139. (2020)
5. Tian, Q., Han, D., Jiang, Y.: Hierarchical Authority Based Weighted Attribute Encryption Scheme. Computer Science and Information Systems,Vol. 16, No. 3, 797-813. (2019)
6. Sharafi, J., Khedmati, Y., Shabani, M.M.:Image Steganography Based On A New Hybrid Chaos Map And Discrete Transforms. Optik, Vol. 226, No. 2. (2021)
7. Rustad, S., Setiadi, D., Syukur, A., Andono, P.:Inverted LSB Image Steganography Using Adaptive Pattern To Improve Imperceptibility. Journal of King Saud University - Computer and Information Sciences, Early Access. (2021)
8. Roy, R.,Sarkar, A., Changder, S.:Chaos based Edge Adaptive Image Steganography. Procedia Technology,Vol. 10, 138-146. (2013)
9. Cataltas, O., Tutuncu, K.: Improvement Of LSB Based Image Steganography. International Journal Of Electrical, Electronics And Data Communication, Vol. 5, 1-5. (2017)
10. Rajput, V., Tiwari, S.K., Gupta, R.: An Enhanced Image Security Using Improved RSA Cryptography And Spatial Orientation Tree Compression Method. International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES).Paralakhemundi, India, 327-331. (2016)
11. Chen, S.K.: A Module-Based LSB Substitution Method With Lossless Secret Data Compression. Computer Standards & Interfaces,Vol. 33, 367-371. (2011)
12. Akhtar, N., Ahamad, V., Javed, H.: A Compressed LSB Steganography Method. 3rd International Conference on Computational Intelligence & Communication Technology (CICT).Ghaziabad, India, 1-7. (2017)
13. Chikouche,S.L., Chikouche,N.: An Improved Approach For Lsb-Based Image Steganography Using AES Algorithm. 5th International Conference on Electrical Engineering - Boumerdes (ICEE-B). Boumerdes, Algeria, 1-6. (2017)
14. Manjula,Y., Shivakumar,K.B.: Enhanced Secure Image Steganography Using Double Encryption Algorithms. 3rd International Conference on Computing for Sustainable Global Development (INDIACom).New Delhi, India, 705-708. (2016)
15. Kasapbaşi,M.C.: A New Chaotic Image Steganography Technique Based on Huffman Compression of Turkish Texts and Fractal Encryption With Post-Quantum Security. IEEE Access,Vol. 7, 148495-148510. (2019)
16. Rachmawanto,E.H., Amin,R.S., Setiadi,D.I.M., Sari,C.A.: A Performance Analysis StegoCrypt Algorithm based on LSB-AES 128 bit in Various Image Size. International Seminar on Application for Technology of Information and Communication (Isemantic). Semarang, Indonesia, 16-21. (2017)
17. Jain,N., Meshram,S., Dubey,S.: Image Steganography Using LSB and Edge – Detection Technique. International Journal of Soft Computing and Engineering, Vol. 2, 217-222. (2012)
18. Shanmugasundaram,S., Lourdusamy,R.: A Comparative Study Of Text Compression Algorithms. International Journal of Wisdom Based Computing, Vol. 1, 68-76. (2011)
19. Langdon,G.G., Rissanen,J.: Compression of Black-White Images with Arithmetic Coding. IEEE Transactions on Communications, Vol. 29, No. 6, 858-867.(1981)
20. Langdon,G.G.: An Introduction to Arithmetic Coding. IBM Journal of Research and Development, Vol. 28, No. 2, 135-149. (1984)
21. Tahghighi,M., Mousavi,M., Khadivi,P.: Hardware Implementation Of A Novel Adaptive Version Of Deflate Compression Algorithm.18th Iranian Conference on Electrical Engineering.Isfahan, Iran, 566-569. (2010)
22. Rivest,R.L., Shamir,A., Adleman,L.: A Method For Obtaining Digital Signatures Ad Public-Key Cryptosystems. Communications of the ACM, Vol. 21, No. 2,120-126.(1977)
23. Xin,Z., Xiaofei,T.: Research And Implementation Of RSA Algorithm For Encryption And Decryption. Proceedings of 2011 6th International Forum on Strategic Technology.Harbin, China, 1118-1121. (2011)

25. Rivest,R.L., Preneel, B. (ed.):The RC5 Encryption Algorithm.Fast Software Encryption. Springer Berlin Heidelberg, Berlin, Heidelberg, 86-96. (1995)
25. Shahzadi,R., Anwar,S.M., Qamar,F., Ali,M., Rodrigues,J.J.P.C.: Chaos Based Enhanced RC5 Algorithm ForSecurity And Integrity Of Clinical Images In Remote Health Monitoring. IEEE Access, Vol. 7, 52858-52870.(2019)
26. McLoone,M., McCanny,J.V.: High-Performance FPGA Implementation OfDES Using A Novel Method For Implementing The Key Schedule. IET Proceedings - Circuits, Devices and Systems,Vol. 150, No. 5, 373-378. (2003)
27. Patidar,V., Sud,K.K.: A Pseudo Random Bit Generator Based on Chaotic Logistic Map and its Statistical Testing. Informatica,Vol. 33, No. 4, 441-452.(2009)
28. Kadhim,I.J., Premaratne, P., Vial,P.J., Halloran,B.: Comprehensive Survey Of Image Steganography: Techniques, Evaluations, And Trends In Future Research. Neurocomputing,Vol. 335, 299-326. (2019)
29. Zhou,W., Bovik,A.C.: A Universal Image Quality Index. IEEE Signal Processing Letters,Vol. 9, No. 3, 81-84. (2002)

Kemal Tütüncü was born in Konya, Turkey, in 1975. He received the master’s degree from Free University of Brussel, Belgium. He received Ph.D. degrees from Selcuk University, Turkey. His research interest includescryptology, information security, natural language processing, and artificial intelligence.

Özcan Çataltaş was born in Konya, Turkey, in 1992. He received the master’s degree from Selcuk University, Turkey. His research interest includes information security and artificial intelligence.

Received: February 27, 2021; Accepted: July 05, 2021.