Improved Deep Hiding/Extraction Algorithm to Enhance the Payload Capacity and Security Level of Hidden Information

Marwa Ahmad[1], Nameer N. El-Emam[2] and Ali F. AL-Azawi[3]

[1,2,3] Department of Computer Science, Philadelphia University, Jordan

***Abstract*:** Steganography algorithms have become a significant technique for preventing illegal users from obtaining secret data. In this paper, a deep hiding/extraction algorithm has been improved (IDHEA) to hide a secret message in color images. The proposed algorithm has been applied to enhance the payload capacity and reduce the time complexity. Modified LSB (MLSB) is based on disseminating secret data randomly on a Cover-Image and has been proposed to replace a number of bits per byte (Nbpb), up to 4 bits, to increase payload capacity and make it difficult to access the hiding data. The number of levels of the IDHEA algorithm has been specified randomly; each level uses a color image, and from one level to the next, the image size is expanded, where this algorithm starts with a small size of a Cover-Image and increases the size of the image gradually or suddenly at the next level, according to an enlargement ratio. Lossless image compression based on the run-length encoding algorithm and Gzip has been applied to enable the size of the data that is hiding at the next level, and data encryption using the Advanced Encryption Standard algorithm (AES) has been introduced at each level to enhance the security level. Thus, the effectiveness of the proposed IDHEA algorithm has been measured at the last level, and the performance of the proposed hiding algorithm has been checked by many statistical and visual measures in terms of the embedding capacity and imperceptibility. Comparisons between the proposed approach and previous work have been implemented; it appears that the intended approach is better than the previously modified LSB algorithms, and it works against visual and statistical attacks with excellent performance achieved by using the detection error (PE). Furthermore, the results confirmed that the Stego-Image with high imperceptibility has reached even a payload capacity that is large and replaces twelve bits per pixel (12-bpp). Moreover, testing is confirmed in that the proposed algorithm can embed secret data efficiently with better visual quality.

***Keywords*:** Steganography; Multi-level steganography; Deep hiding/extraction; Least significant bit; High payload; high security.

1. Introduction

Currently, data hiding is one of the most important requirements, where data is sent and received online at every moment, and it requires security when it is shared over the Internet. Hiding information is an active research area, where confidential information is included in a carrier, such as photos and videos, to hide their presence while maintaining their visual quality.

Researchers have provided different techniques for information concealment since the previous decade; they have focused on the load capacity and image quality Al-Shatanawi and El-Emam [1], El-Emam and Al-Zubidy [2], and Muhammad et al., [3]. In general, five main objectives are used to evaluate the performance of data-hiding algorithms, which include the embedding capacity, imperceptibility, security, robustness and complexity Darabkh et al. [4]. Recently, many analytical techniques have been developed to extract significant hidden information from Stego-Image s. Therefore, to avoid data extraction, several new steganography algorithms were improved by many researchers to make it difficult for the human visual system (HVS) to observe the difference between the stego- and Cover-Image s. In addition, to increase the security level and payload capacity, a multi-level steganography technique (MLS) was proposed by Sikarwar [5]; see Figure 1.

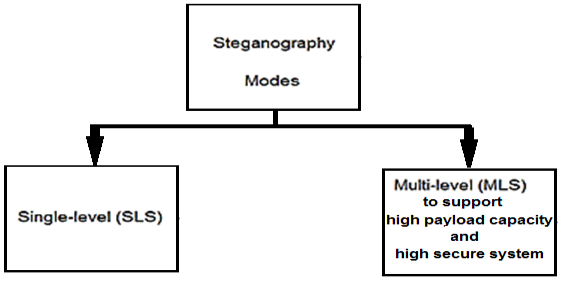


Figure 1. Modes of Steganography

The rest of the paper is structured as follows: In section 2, related works, and the proposed steganography algorithm specifications has appeared in section 3. In section 4 the requirements of improved deep hiding/extracting algorithm with the algorithm design and its implementation are presented in section 4. In section 5. The experimental results are discussed in. Finally, in section 6 the main conclusions of the proposed algorithm have been discussed.

2. Related Work

Steganography is necessary due to the exponential growth and secret information of possible computer users over the Internet, so it is a technique of invisible information to keep secret data text, audio, and video files inside other data. Steganalysis is the technology that attempts to identify and extract the hidden data.

Siswanto1 et. al [6] discussed that the performance of the encryption algorithm is low when considering the encryption speed. On the other hand, a simple encryption algorithm will have faster encryption speed; however, it generally provides low-security protection.

Different techniques that use multiple bit replacement per pixel were discussed by El-Emam [7], El-Emam, and Al-Diabat [8], and El-Emam and Qaddoum [9]. These techniques are based on a modified least significant bit (MLSB) algorithm to work against the steganalysis process. The MLSB methods are precipitated by the notion that it is possible to detect the hidden data by using statistical analysis, such as the chi-square test or K-S test Ker [10].

Latika and Gulati [11] explained the types of nonlinear media that are used in data hiding techniques and classified these types of media into four main classes: text, image, audio, and

video. The first three of these classes hide the data in the same file format. The protocol is much advanced relative to the other classes and strives to use the platform or protocol to hide data. Belhamra and Souidi [12] defined steganography over Redundant Residue Number System (RRNS) Codes and explained distortion-less RRNS based steganographic techniques with an investigation of their corresponding hiding capacities and then discussed their results. Sikarwar [5] proposed another approach based on an integrated and synchronized protocol, connecting the applications of the secure algorithm, powerful key cryptography, and multilevel steganography since it is more effective compared to simple steganography. Multilevel steganography and powerful cryptography derive the idea about a combined synchronized protocol for secure data transmission.

Jain and Kumar [13] applied the technique based on the substitution method using the least significant bit (LSB); as is known, the LSB has a wide range of applications with the ability to hide information in 8-bit and 24-bit images. This method works on the concept that data is hidden in the least significant bit of a pixel byte, thereby making it essentially impossible for the human eye to discern that there is a difference between the new image, which has the secret data, and the old image.

There are several methods for hiding data within pixels, the restricting factor is always the number of bits that are replaced in each pixel. Therefore, the main contributions of this paper aim to reach equilibrium between the amount of hiding data and the level of acceptable perversion, as well as granting a high level of protection.

Least-Significant Bit (LSB) hiding is the simplest method in steganography. It is imperceptible to the human eye, and it is very reasonable to yield to statistical analyses (Steganalysis publication). Darabkh, A, et al., (2017) [4] proposed a steganography algorithm based on multi-directional of pixel-value differencing (MDPVD) and modified LSB. Akbay et al. [14] used the least significant bit (LSB) modified method with a shuffle algorithm to hide data in color images in 24-bit JPEG format. Abdulwahed [15] used the hyper technique to compress the secret data by using different layers of security working together to increase protection from attacks. Sahu and Swain [16] suggested an improved image steganographic technique based on the modified least significant bit (LSB) substitution and LSB matching, to improve payload capacity and peak signal-to-noise ratio (PSNR).

Many challenges are faced in the process of data hiding in images, such as data protection, image quality, hiding capacity, and computational complexity. To maintain these challenges, a data hiding method for color images using the benefits of the imperceptibility feature of the LSB technique working in the spatial domain with robust hiding methods based on salient features guided by human visual perception extracted from the transform domain. Swain [17] proposed a steganography technique by dividing the image into non-overlapped pixel blocks. For every pixel, the least significant bit (LSB) is employed on two LSBs, and quotient value differencing (QVD) is implemented on the resting six bits. Nipanikar and Deepthi [18] proposed a method of hiding the text message in the image using a Discrete Wavelet Transform (DWT) with a cost function to locate a position to support hiding data. The cost function uses entropy, intensity, and the edge of the image to calculate the position of hiding.

A secret image sharing scheme with identity-based authentication is proposed by Wang et al. [19], where an image is encoded, and compressed, then the compressive image is divided into n shadows according to an innovative way in which the production of shadows is managed by the associates' identification number.

Moreover, Steganography concerns the following areas: confidential communication and secret data storing, protection of data alteration, access control system for digital content distribution, and media database systems). Hashim et al., [20] constructed a steganography system based on two control parameters selected randomly and multi-level encryption, to ensure the imperceptibility of the Stego-Image. Two-levels of data security are called multilevel steganography, and this approach was conducted by Tyagi, and Anurag [21], one level is image steganography, and the other level is video steganography, where the text is encrypted by using a digital encryption system (DES). The encrypted data is embedded in the image by using the LSB technique. Kasana et al. [22] proposed a block-based steganography approach. The Cover-Image is factorized into parts of equal size, and the largest pixel of each part is found to embed the secret data and the smallest pixel of each part is used for hiding, where hiding of secret data is performed using the multilevel approach. Xue et al. [23] proposed a multi-layer steganographic method using the collaboration of (MLS) and audio time-domain segmented steganography (ATDSS) and Network steganography (NS). MLS-ATDSS&NS is realized in two covert layers (audio steganography layer and network steganography layer) by two steps. Adeniji et al. [24] proposed a multi-level steganography system to hide data transmission and participating over the Internet, where cryptographic and steganographic methods were combined. This study also proposed compressing the secret file using the LZW algorithm. Elshare and EL-Emam [25] proposed a steganography algorithm to hide as much secret data as possible in color and grey images; this algorithm applied a deep hiding and extraction algorithm (DHEA), and it is also based on the modified multi-level steganography (MLS). Bhowal [26] presents multilevel audio steganography, which represents a model for hidden communication in communication technology. At least two embedding methods are used in such a way that the second method will use the first method as a carrier.

3. The Proposed Steganography Algorithm Specifications

The proposed steganography algorithm to conceal a large size secret message (Sm) has been applied effectively by enhancing the multi-level steganography (MLS) technique. The previous work proposed by Elshare and EL-Emam [25] was based on MLS, and it is characterized by using a deep hiding (DHD) algorithm to hide a secret message at one level selected randomly from a uniform size of deep levels. In this work, we modified the previous approach proposed by Elshare and EL-Emam [25] by distributing a secret message on many levels to enhance the payload capacity, and we used a non-uniform size between levels to make it difficult to detect the secret message.

The main specifications suggested in this work are described through the following definitions:

Definition 1: The Stego-Image (SI^L) at the first level (L=1) is generated by replacing bits of a secret message of the first level Sm^1 with bits of a Cover-Image (CI^1) at the first level using a specific cipher key (CK) and specific hiding algorithm.

Definition 2: The Stego-Image (SI^L) at the other levels (L>1) is generated by replacing bits of secret message (Sm^L) of the L^{th} level and the Stego-Image (SI^{L-1}) of the $(L-1)^{th}$ level with bits of a Cover-Image (CI^L) of the L^{th} level using a specific cipher key (CK) and specific hiding algorithm.

Definition 3: Let HD1L be a hiding function at the first level (L=1) as defined in the following map:

$$\boldsymbol{HD1^L}: CI^L \times ECSm^L \times CK \times \left[loc_{c,i}^L\right] \rightarrow SI^L$$

where the function domain has four parameters (CI, ECSm, CK, $\left[\text{loc}_{c,i}^{L}\right]$) such that ECSm is a compressed and encrypted secret message, and $\left[\text{loc}_{c,i}^{L}\right]$ is a set of locations in a Cover-Image (CI^L) at level (L) for a specific color (c), where $c \in \{R, G, B\}$ and where these locations are used for hiding a secret message.

Here, $HD2^L$ is a hiding function at the other levels (L>1) and is defined in the following map:

$$\boldsymbol{HD2^L}: CI^L \times ECSm^L \times CK \times \left[loc_{c,i}^L\right] \times SI^{L-1} \rightarrow SI^L$$

where the function domain has five parameters (CI^L, $ECSm^L$, CK, $\left[\text{loc}_{c,i}^{L}\right]$, SI^{L-1})

Definition 4: Let PL be a Payload capacity of hidden information such that $= |Sm|$.

Definition 5: Let $EX1^L$ be an extracting function at the last level (L=n) defined in the map:

$$\boldsymbol{EX1^L}: SI^L \times CK \times \left[loc_{c,i}^L\right] \rightarrow SI^{L-1}$$

where the function domain has three parameters (SI^L, CK, $\left[\text{loc}_{c,i}^{L}\right]$), whereas the extraction function ($EX2^L$) at the other levels (L< n) is defined in the following map:

$$\boldsymbol{EX2^L}: SI^L \times CK \times \left[loc_{c,i}^L\right] \rightarrow < SI^{L-1} \times ECSm^L >$$

where the function domain has three parameters (SI^L, CK, $\left[\text{loc}_{c,i}^{L}\right]$), and the range has two parameters (SI^{L-1}, $ECSm^L$).

Definition 6: Let IDHD be an improved deep hiding function (call n-times hiding function) defined in the following map:

$$\boldsymbol{IDHD}: (HD1^L)^1 \times (HD2^L)^{n-1} \rightarrow SI^{\hat{L}}$$

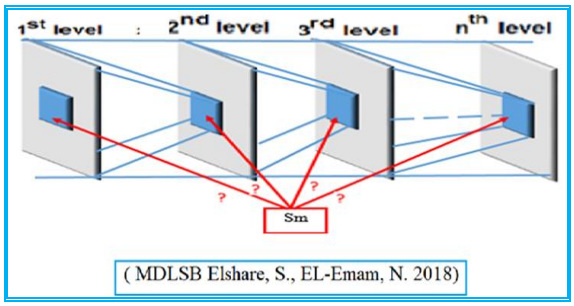
where $\hat{L}$ is the last level (level n), and L is all levels less than n.

Definition 7: Let IDEX be an improved deep extraction function (call n-times hiding function) defined in the following map:

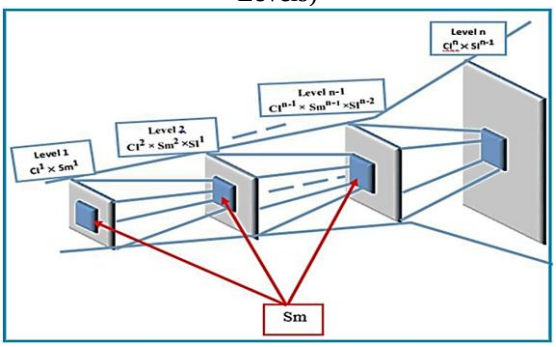
$$\boldsymbol{IDEX}: (EX2^L)^1 \times (EX2^L)^{n-1} \rightarrow < CI^1 \times ECSm^1 >$$

4. Improved Deep Hiding/Extraction Algorithm

The improved deep hiding/extraction algorithm (IDHEA) is introduced in this paper; it is based on multi-level steganography (MLS) to enhance the data security and payload capacity.



Deep hiding using a fixed size of images (Uniform Levels)



Improved deep hiding using a variable size of images (Gradual or Sudden Enlargement) - Proposed

Figure 2a. Improved Deep Hiding Based on MLS

The proposed algorithm IDHEA starts with a small size of the Cover-Image and increases the size of the Cover-Image from one level to the next gradually or suddenly according to the enlargement ratio (see Figure 2a). This new approach is efficient by reducing the time of the hiding process. The proposed (IDHEA) algorithm has been implemented by using the hiding and extraction phases.

The software development of each phase is defined, and it includes three steps: system requirements, model design, and implementation. We observed that when the angle degree (α1) is small, a gradual enlargement of the levels has appeared, and then, deep hiding (long length L1) can be applied, whereas semi-deep hiding (short length L2) is applied when a sudden enlargement of the levels has arisen with a large degree of angle (α2); see Figures (2b, 2c). Moreover, we see that applying a gradual or sudden enlargement of the levels can reduce the execution time by 50% instead of using uniform levels based on deep hiding.

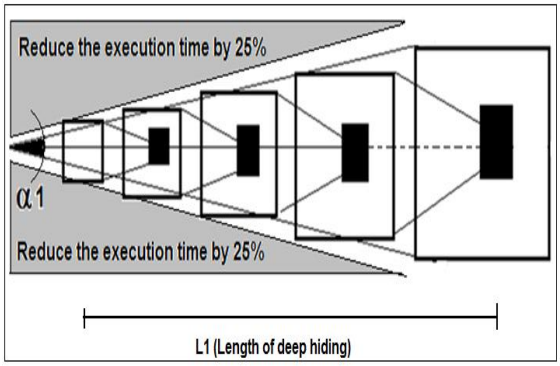


Figure 2b. Gradual enlargement of levels based on deep hiding using a small degree angle (α1)

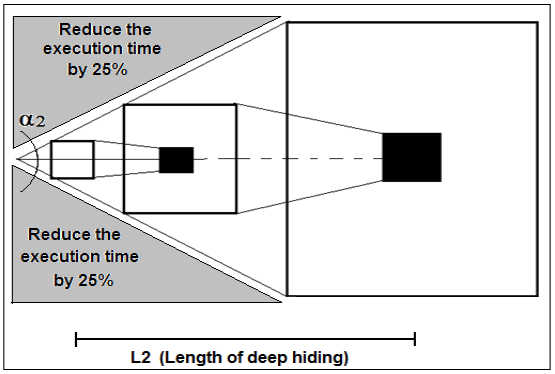


Figure 2c. Sudden enlargement of levels based on semi-deep hiding using a large degree angle (α2)

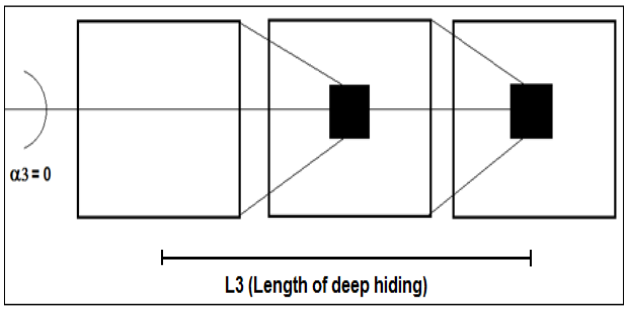


Figure 2d. Uniform levels based on deep hiding using a zero-degree angle (α3=0) Elshare and EL-Emam [25]

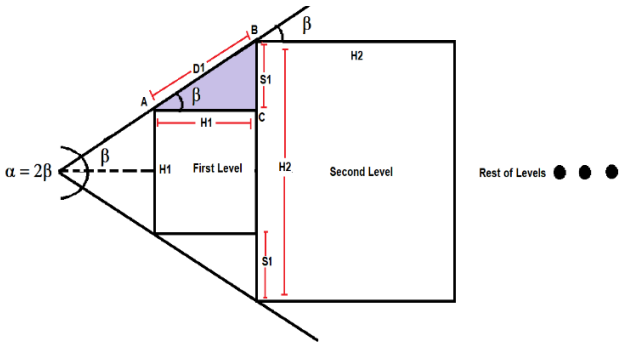


Figure 2e. The analysis side to find the number of levels

4.1 Calculating the number of levels

According to Figure 2e, we can calculate the number of levels and the size of cover image at each level according to Algorithm1:

Algorithm 1: Length_ of_ levels (.){
Input α, where α is an angle (in degree) to find the enlargement ratio of the size of cover image CI through a sequence of levels;
Calculate the angle β, where β= α/2;
Define the initial size of cover image CI^1 at the first level ($H^1 \times H^1$);
For (int i=1, ∞) { // where *i is a level's index*
Compute the Hypotenuse (D^i) of the triangle using the formula:

$$D^i = \frac{H^i}{Cos\ (\beta)} \qquad (1)$$

Compute the Perpendicular (S^i) of the triangle using the formula based on Pythagoras theorem:

$$S^i = \sqrt{\left(\frac{H^i}{Cos\ (\beta)}\right)^2 - (H^i)^2} \qquad (2)$$

Compute the size of the next cover image CI^{i+1} at the next level ($H^{i+1} \times H^{i+1}$);

$$H^{i+1} = H^i + (2 \times S^i) \qquad (3)$$

IF ($H^{i+1} > 1024$)
Break; // *Exit the loop to avoid the overflow of the size of a Cover Image CI*
} //*end for i*
} // *end Algorithm 1*

4.2 System requirements

The Advanced Encryption Standard (AES) encryption/decryption tools have been implemented to encrypt/decrypt secret data before the hiding/extraction process Castillo et al. [27]. Moreover, lossless compress/decompress algorithms called the run-length encoding algorithm and Gzip have been introduced to compress/decompress data that are used through levels of hiding/extraction algorithms. In addition, the proposed improved deep hiding/extraction software have been constructed under the Visual Studio 2019 C# program and using MATLAB version R2015a to represent the results.

4.3 Model design

The IDHEA processes include two stages; these are the hiding and extraction stages.

4.3.1 Hiding stage

The hiding stage has been implemented on a wide range of secret information and maintain s a security level of hidden information from attackers while it is subject to both visual and statistical attacks.
The proposed approach improves deep hiding of information, which is based on multi-level steganography (MLS). In the proposed approach, a secret message (Sm) has been distributed over multiple levels to increase the security and payload capacity and to avoid degradation of the Stego-Image for each level. For each level, we generate a new cipher key to produce a non-uniform hiding approach among levels; this approach is useful for making it difficult to detect a secret message.
For each level, a sequence of procedures has been implemented to hide secret data in a Cover-Image, where the first procedure is based on data compression of a secret message (Sm) and Stego-Image (SI) to produce (CSm) and (CSI), respectively. The second procedure concentrates on the encryption of (CSm) and (CSI) to produce (ECSm) and (ECSI), respectively, while the third procedure is the hiding technique to embed randomly (ECSm) in a Cover-Image (CI). In this paper, a deep hiding technique has been improved by introducing a modified least significant bit (LSB) and hiding data randomly by using a non-uniform cipher key (CK) among the levels except for the last level, which requires an extra process to confirm the imperceptibility of the secret data; this goal is reached by scattering the secret data randomly and hiding at most two bits in each byte to confuse the attackers. The hiding technique must prepare a Cover-Image (CI) that holds compressed and encrypted secret data, where the output of the i^{th} level is the i^{th} Stego-Image (SI^i), which is used as the input to the $(i+1)^{th}$ level in addition to the $(i+1)^{th}$ secret message (Sm^{i+1}). The dynamic process has been adapted from one level to the next level by varying the method of calculating the cipher key (CK) and applying different data hiding techniques among the levels.

Figure 3 illustrates the hiding phase to obtain deep hiding of the data using N levels. The random pixels selection algorithm has been constructed to hide a secret massage Sm in the Cover-Image CI; this approach is applied to increase the security and to work against the attackers. The set of pixel locations has been generated using a random function based on an input seed value, where the set of locations has been used on three colors {R, G, and B}.
The proposed hiding algorithm based on the modified least significant bit (MLSB) has two phases to define the number of bits per byte (Nbpb) that are replaced in a Cover-Image.

Phase 1:-For each level except for the last level (level n), Nbpb =4
Phase 2: - For the last level, Nbpb = {1 or 2 or 4} according to specific conditions; see Eq. (3).

The Improved Deep Hiding Algorithm (IDHA) is defined in algorithm 2 to algorithm 5 as follows:

Algorithm 2: Image IDHA (.) // Improved Deep Hiding Algorithm.
/*
Let us define the following:
- n be the number of levels; // *using Algorithm1*

- L be the level;
- Sm be the secret message;
- SI be the Stego-Image;
- CSm be the compressed secret message;
- ECSm be the encrypted and compressed secret message;
- len be the length to encrypt a compressed secret message;
- CSI be the compressed Stego-Image;
- ECSI be the encrypted compressed Stego-Image;
- $\left[\text{Loc}_{X,i}^{L}\right]$ be the set of locations at level L with color x, where x∈{R, G, B};
- CK be the Cipher key;
- CI be the Cover-Image;

*/
Input n;
Input $CI^1, CI^2, \ldots, CI^n$;
// *where* $|CI^L| < |CI^{L+1}|$, $\forall L$, *such that* $|CI^{L+1}| = \alpha^L |CI^L|$ *and* α^L *is the enlargement ratio at level L* $\forall L=1,\ldots,n-1$, *and* α^n *is the enlargement ratio at level n,such that* $\alpha^n \in [4,\infty]$, *and it is better to make* α^n *large noughto redo the noise.*
Input Sm;
Extract the set of sub-secret messages {$Sm^1, Sm^2, Sm^3 \ldots Sm^{n-1}$}, such that concatenation of the sub-secret messages is defined in Eq. (4):

$$Sm = \bigoplus_{L=1}^{n-1} Sm^L \quad (4)$$

// *where the length of* Sm^L *at level L is calculated in Eq. (5).*

$$|Sm^L| = \beta^L \times |CI^L| \quad (5)$$

// *where* $\beta^L \in [0, 0.5]$ $\forall L= 1,\ldots, n-1$
For (L=1, n) {
Input W^L; // *Width of* CI^L
Input H^L; // *Height of* CI^L
Input CK^L;
Find CSm^1 =**Compress** (Sm^1); // *Compress* Sm^L;
Find $ECSm^1$ =**Encrypt** (CSm^1); // *Encrypt* CSm^L;
Find len = | $ECSm^L$ |; // *Length of* $ECSm^L$;
Compute $seed^L$= **int**[$((W^L)^2 + (H^L)^2)^{0.5}$]; // *find seed to generate random location*
Find set of location $\left[\text{loc}_{c,i}^{1}\right] = \{\text{loc}_1, \text{loc}_2, \ldots, \text{loc}_{HxW}\}$ by using Rand_Loc (seed1, W^L, H^L)
// *Apply IDHD; see definition 6.*
If (L=1) {
For each color c in {R, G, B} {
For (i=1, len/3) { // *to find byte location, we have len/3 locations for each color*
For (j=1, 4) { // *to move to next bit j*
$SI^1 \oplus = HD1^1$ (CI^1, $\text{loc}_{c,i}^{1}$ (j), $ECSm^1{}_c$, CK^1);
} // *End for j*
} // *End for i*
Find CSI^L = **Compress** (SI^L); // *using compress Gzip*
} // *End For each color*
} // *End if*
Else if (L>1 and L<n) {
Find CSm^L =**Compress** (Sm^L); // *Compress* Sm^L;
Find $ECSm^L$ =**Encrypt** (CSm^L); // *Encrypt* CSm^L;
Find len = | $ECSm^L$ |; // *Length of* $ECSm^L$;
Compute $seed^L$= **int**[$((W^L)^2 + (H^L)^2)^{0.5}$]; // *find seed to generate random location*
Find set of locations:
$\left[\text{loc}_{c,i}^{L}\right] = \{\text{loc}_1, \text{loc}_2, \ldots, \text{loc}_{HxW}\}$
// *by using Rand_Loc (seed1,* W^L, H^L*)*

For each color c in {R, G, B} {
For (i=1, len/3) { // *to find the byte location, we have len/3 locations for each color*
For (j=1,4) {// *to move to the next bit*
$SI^L \oplus = HD2^L$ (CI^L, $\text{loc}_{c,i}^{L}$ (j), $ECSm_c{}^L$, CSI^{L-1} , CK^L);
} // *End for j*
} // *End for i*
Find CSI^L = **Compress** (SI^L); // *using compress Gzip,*
} // *End for each color*
} // *End else..if*
Else if (L=n) {
Find CSm^L =**Compress** (Sm^L); // *Compress* Sm^L;
Find $ECSm^L$ =**Encrypt** (CSm^L); // *Encrypt* CSm^L;
Find len = | $ECSm^L$ |; // *Length of* $ECSm^L$;
Compute $seed^L$= **int**[$((W^L)^2 + (H^L)^2)^{0.5}$]; // *find seed to generate the random location*
Find set of locations:
$\left[\text{Loc}_{c,i}^{L}\right] = \{\text{loc}_1, \text{loc}_2, \ldots, \text{loc}_{HxW}\}$
// *by using Rand_Loc (seed1,* W^L, H^L*)*
For each color c in {R, G, B} {
For (i=1, len/3) { // *to find the byte location, we have len/3 locations for each color*
Appy modified LSB (MLSB) according to the following:
Find Nbpbi at the i^{th} location using the standard deviation and mean of 4- neighbours; see Eq. (6).

$$Nbpbi = \begin{cases} 1 & if\ \alpha^n = 4\ and\ \sigma^{4-Neighbours} < 10^{-5} \\ 2 & if\ \alpha^n = 4\ and\ \sigma^{4-Neighbours} = 10^{-5} \\ 3 & if\ \alpha^n = 4\ and\ \sigma^{4-Neighbours} > 10^{-5} \\ 4 & if\ \alpha^n > 4 \end{cases} \quad (6)$$

// *where* $\sigma^{4-Neighbours}$ *is defined in Eq. (7).*

$$\sigma^{4-Neighbours} = \left(\sqrt{\frac{1}{4}\sum_{i=1}^{4} (CI_i - \mu_i{}^{4-Neighbours})^2}\right) \quad (7)$$

// and $\mu_i{}^{4-Neighbours}$ *is the mean of (4 – neighbours) aroud the i[th] location.*

For (j=1, $Nbpb_i$) { // *to move to the next bit*
$ECSm_c{}^n$ = " "; | $ECSm_c{}^n$|=0 in the last level
$SI^n \oplus = HD2^L(CI^n, loc_{c,i}^n (j), ECSm_c{}^n, CSI^{n-1}, CK^L)$;
} // *End for j*
} // *End for i*

Find CSI^n = Compress (SI^n); *// using compress Gzip,*
} // *End For each color*
} // *End else..if*
} // *End for L*
//*End of **Algorithm 2***

Algorithm 3: Image HD1 (CI, Loc, Sm, CK) { // *Data hiding*
Find bit value bv from Sm;
Replace one bit of a cover bit at the location loc by the bit bv;
} // *End of **Algorithm3***

Algorithm 4: Image HD2 (CI, Loc, Sm , SI, CK) { // *Data hiding*
Find bit value (bv1) from Sm;
Replace one bit of a Cover-Image bit at the location loc by the bit bv1;
Find bit value (bv2) from SI;
Replace one bit of a Cover-Image by the bit at the location (loc+1) by the bit bv2;
} // *End of **Algorithm4***

Algorithm 5: int [] Rand_Loc (seed, W, H) { // *Find a random stream for the bytes' locations*
Make a permutation for seed, 2*seed and 3*seed to obtain the random locations for the red, green, and blue colors by the following:

int max = W*H;
// *Create a one-dimensional array Loc with length=max,*
For (i=0 ,..,max) {
Loc(i)=i;
// *Create a one-dimensional array called index with a size equal to max.*
index (i)= Rnd(seed);
} // *end for i*

For (i =0,…,.max) {
// *make a swap between the Loc array elements*
int temp = Loc(i)
Loc (i) = Loc(index(i))
Loc(index(i)) = temp;
} // *end for i*
Return set of location {Loc(1),…, Loc(max) {
} // *End of **Algorithm 5;***

The time complexity measures are studied for the proposed (IDHA) and are defined in Eq. (8).

$$Time\ (IDHA)\ \approx\ O\ (n^2) \quad (8)$$

4.3.2 Extraction stage

In this stage, we improved the deep extraction algorithms (IDEAs), whereby after the recipient receives the Stego-Image, the agreed parameters have been entered for each level; these parameters are (seed, number of levels and cipher key). These parameters have been used to extract Sm and SI from each level and then decrypt, decompress, and obtain the next Sm and SI. The steps are applied until the required confidential message is combined (see Figure 4).

Algorithm 6: Image IDE (.) // Extracting
/*
Let us define the following:
- n be the number of levels;
- L be the level number;
- Sm be the secret message;
- SI be the Stego-Image ;
- DCSm be the decompressing secret message;
- DEDCSm be the decrypted decompressed secret message;
- len be length to decrypt a decompressed secret message;
- DCSI be the decompressed Stego-Image ;
- $LOC_{X,i}^L$ be the set of locations at the specific color x, where x∈{R, G, B}
- CK be the cipher key;

*/
For (L= n,1) { // *decrement the value of L*
Input CKL;
If (L= n) {
For each color c in {R, G, B} {
For (i=1, len/3) { // *to find byte location, we have len/3 locations for each color*
Input $seed_i$; // *find seed to generate the random locations*
Find Loc_i = Rand_Loc ($seed_i$) // *find locations that are used for hiding the secret data*
Appy modified LSB (MLSB) according to the following:
Find $Nbpb_i$ at the i^{th} location using the standard deviation σ and mean of 4- neighbours according to Eq. (6)

$$Loc_i^n = Loc_i^n + 1$$

$$DCSI^{n-1} \oplus = \mathbf{EX1^n}(SI^n, Loc_i^n, CK^n);$$

} // *End for i*

Find SI^{n-1} = **Dcompress (**$DCSI^{n-1}$); // *Dencrypt $CSBSm_1$;*
} // *end for each color*
} // *End else..if*
Else If (L < n) **{**
For each color c in {R, G, B} {
For (i=1, len/3) { // *to find the byte location, we have len/3 locations for each color*
Input $seed_i$; // *find seed to generate the random location*

Find set of lacations:
$locations[loc_{c,i}^L] = \{loc_1, loc_2, \ldots, loc_{HxW}\}$
// *by using Rand_Loc (seed1, W^L, H^L) to find locations that are used for hiding secret data*
For (j=1, 4) { // *to find bit location*

$< DCSI^L, DEDCSm^L > \oplus = EX2^L (Loc^L_{c,i} (j), CSI^{L-1})$;
} // *End for j* .
} // *End for i.*
Find SI^L = **Dcompress (**$DCSI^L$); // *decompress DCSI* ^{L}*;*
Find $DCSm^L$ = **Dencrypt (**$DEDCSm^L$); // *dencrypt* $DEDCSm^L$*;*
Find Sm^L = **Dcompress (**$DCSm^L$); // *decompress* $DCSm^L$*;*
} // *End For each color*
} // *End if*
} // *end for L*
// *End* ***Algorithm 6***

Algorithm 7: **Image EX1** (SI, Loc, CK) {
// *Data hiding*
Extract one bit of a SI^{L-1} from the location loc of SI^L;
} // *End of* ***Algorithm 7***

Algorithm 8: **Image EX2** (SI, Loc, CK) {
// *Data hiding*
Extract one bit of a SI^{L-1} from the location loc of SI^L;
Extract one bit of a Sm^L from the location loc+1 of SI^L;
} // *End of* ***Algorithm 8***

The time analysis and time complexity measures are studied in the proposed (IDEA) and are defined in Eq. (9).

$$Time\ (IDE) \approx O\ (n^2) \qquad (9)$$

4.4 Implementation of improved deep hiding algorithm (IDHA)

Assume that we have one byte from SmL at the (Lth)-level and one byte from SIL-1 at the ((L-1) th) -level and that we need to hide those bytes in CIL at the (Lth)-level. In Figure 5, we explain step-by-step the implementation process of the hiding technique.

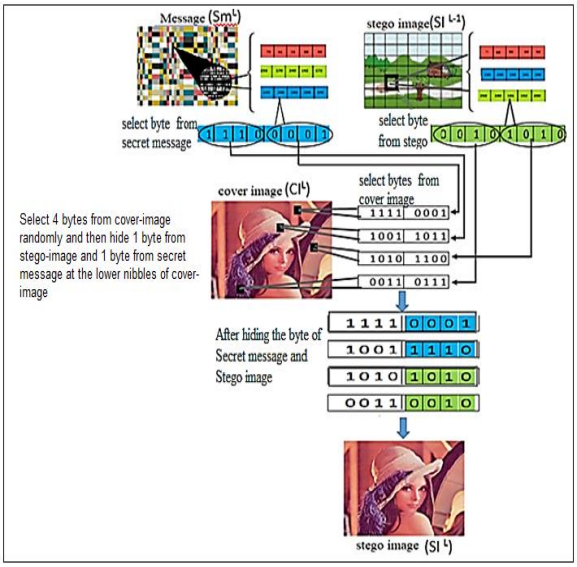


Figure 5. Implementation of the hiding process

5. Results and Discussion

The empirical results have been presented in this study by selecting more than 500 color images from the (UCID v2) database; these images are used for testing the proposed technique. In this section, we present the results of six color images and make a comparison with the other techniques; see Figure 6. The dataset of the Cover-Image s includes different sizes of images from a small scale (150 x150) to a large scale (1080 x 1024). This distinction is needed to examine the effectiveness of the data hiding and the mass of the payload capacity for each measure.

The number of colors in the images is a necessary constituent in the suggested algorithm to define the number of bits per byte at each mask (4-neighbour bytes) that has the smallest effect over the whole image. Furthermore, the standard deviation of each mask at each color component is calculated to define the data hiding capacity. Accordingly, the dataset involves different colors intensities to cover all of the details of the hiding process. The image resolution factor plays the main part in the proposed algorithm. Thus, the high resolution of the image serves to reduce the noise on the Stego-Image; a resolution factor has been introduced in the experiment to estimate the achievement of the algorithm.

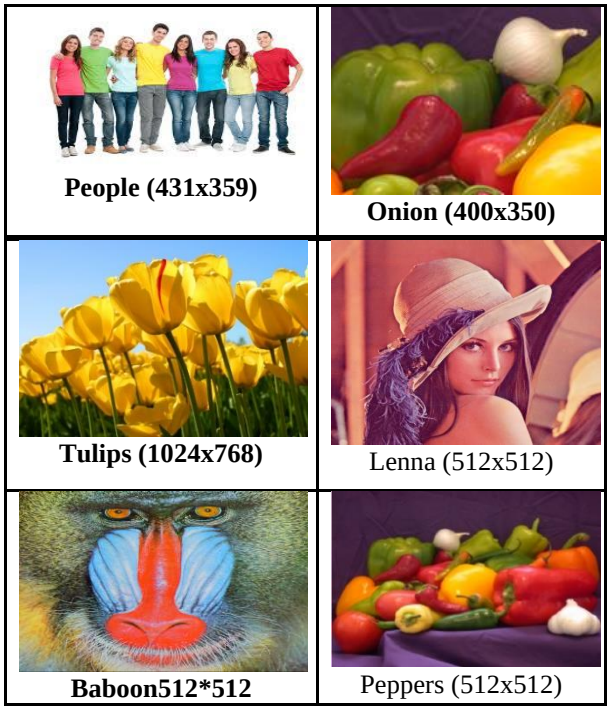


Figure 6. Sample of bitmap images from the (UCID v2) database used for testing the proposed approach

5.1 Results comparisons with the other techniques

A full test is performed over the set of images in the dataset, and thus, the results of the proposed approach have been compared with the other techniques under the same situation. Table 2 shows some of the achieved results based on three metrics (MSE, SNR, and PSNR) defined in Eqs. (10, 11, and 12), respectively.

$$MSE = \frac{1}{h*w}\sum_{i=0}^{h-1}\sum_{j=0}^{w-1}[CI(i,j) - SI(i,j)]^2 \qquad (10)$$

$$SNR = \frac{\mu}{\sigma} \qquad (11)$$

where μ is the signal mean, and σ is the standard deviation of the noise.

$$PSNR = 10\ log_{10}\left(\frac{MAX^2}{MSE}\right) \qquad (12)$$

where the MAX is the maximum possible pixel value of the Stego-Image.

The proposed technique MLSB working under IDHEA has been compared with existing techniques such as MDLSB

working under DHEA Elshare and EL-Emam [25] and the traditional Least Significant Bit (LSB). The results on the three metrics (MSE, SNR, and PSNR) of the proposed techniques are better than those for the other techniques; see Table 1. In addition, the results of the metrics SNR, PSNR of the proposed technique using a large payload capacity are better than LSB & MDLSB by approximately 26.6% and 11.3%, respectively, for the People1 image; the results are better by approximately 13.7% and 34.82%, respectively, for the Tulips image; the results are better by approximately 12.8% and 19.2%, respectively, for the Lena image; and the results are better by approximately 15.3% and 18.2%, respectively, for the Onion image.

Furthermore, we calculate the metric MSE on four Stego-Image s generated by the proposed algorithm, and the results show that the maximum MSE is 1.8527, which appeared on the Onion image due to the small ratio (which was 1.0625:1) between the size of the image and the payload capacity, while the minimum MSE is 0.0687, which appeared on the Tulips image due to the small ratio between the size of the image and the payload capacity, which was (6:1).

Moreover, the measurement of the maximum SNR of the proposed algorithm among the four images is 55.395 dB on the Tulips image and is better than those for MDLSB and LSB, which were approximately 6.75 dB and 24.76 dB, respectively, whereas the minimum SNR of the proposed algorithm among the four images is 38.445 dB on the Onion image, which is better than those for MDLSB and LSB, which were approximately 5.0741 dB and 21.2241 dB, respectively.

Furthermore, the maximum PSNR of the proposed algorithm among the four images is 59.760 dB on the Tulips image and is better than those for MDLSB and LSB, which were approximately 6.07 dB and 24.77 dB respectively, whereas the minimum PSNR of the proposed algorithm among the four images is 45.4528 dB on the Onion image and is better than those for MDLSB and LSB, which were approximately 6.98 dB and 21.223 dB, respectively.

5.2 Comparison of the results with previous work

The effect of the payload capacity to measure the image distortion is presented in this section using the PSNR metric. Table 2 shows the performance of the MLSB that works under the IDHEA algorithm, where results show that the proposed algorithm is better than the MDLSB technique working under the DHEA algorithm by Elshare and EL-Emam [25] , the algorithms proposed by Ou et al. [28], and Li et al. [29], where this study has been implemented over four bitmap test-colored-images with size (512×512) from the (UCID v2) database.

The results show that the PSNR metric of the proposed scheme of the minimum payload capacity (2×104 bits) is better than the other algorithms mentioned above by approximately 7.95%, 10.23%, and 4.76%, respectively, for the Lena image; it is better by approximately 10.33%, 9.86%, and 6.54%, respectively, for the Baboon image; it is better by approximately 4.76%, 8.14%, and 2.91%, respectively, for the Barbara image; and it is better by approximately 8.90%, 10.75%, and 5.66%, respectively, for the Peppers image. However, the PSNR results of the proposed scheme for the maximum payload capacity (15×104 bits) of the Lena image is better than the first and the second algorithms mentioned above by approximately 10.95% and 4.66%, respectively. The PSNR results of the proposed scheme for the maximum payload capacity (5.6×104) bits of the Baboon image is better than that for the other algorithms mentioned above by approximately 7.07%, 6.88%, and 0.35%, respectively. The PSNR results of the proposed scheme for the maximum payload capacity (12.5×104 bits) for the Barbara image is better than that for the other algorithms by approximately 7.21%, 7.21%, and 0.49%, respectively. The PSNR results of the proposed scheme for the maximum payload capacity (10.5×104 bits) for the Peppers image is better than that for the first and the second algorithms mentioned above by approximately 10.35% and 2.15%, respectively.

The proposed deep hiding mechanism has been implemented to reduce the probability of data extraction by (n-1)/n, where n is the number of levels. Therefore, this approach leads to reach a higher level of security over other proposed techniques based on single level steganography (SLS).

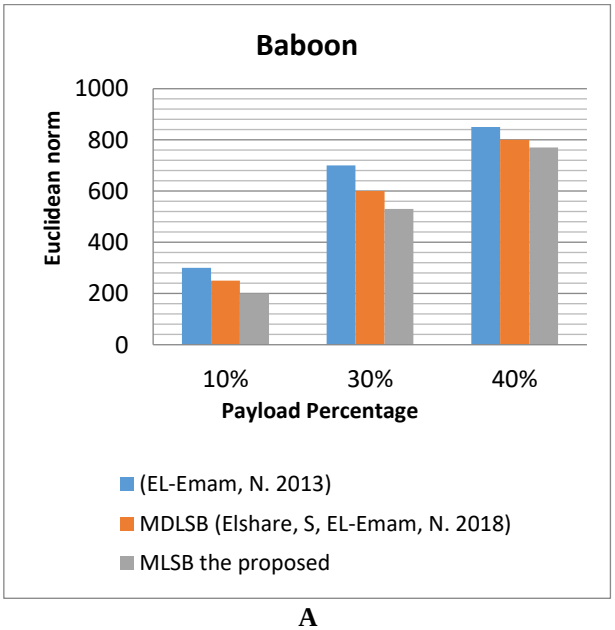
The experimental results illustrated in Table 2 shows that the PSNRs of the selected test images using the proposed hiding algorithm IDHEA are greater than the PSNRs of the existing algorithms of MDLSB under DHEA Elshare and EL-Emam [25], Ou et al., [28], and Li et al. [29]. The proposed approach leads to a higher level of security if the number of levels (n) is large. Another major enhancement in the proposed algorithm IDHEA is to apply the non-uniform hiding criteria at each color pixel. This criterion is based on the standard deviation (σ) at each mask (4 neighbour bytes).

5.3 Euclidean norm

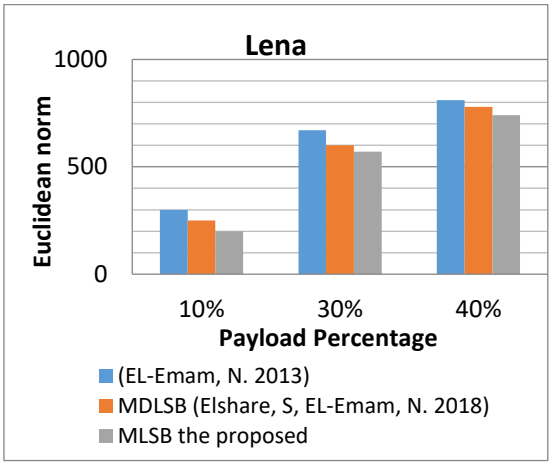
The Euclidean norm test in Eq. (13) has been implemented on three-color images with the size (512 X 512) to check the distance (d) between the Cover-Image and Stego-Image for the three color components {R, G, B}.

$$d = \sqrt{(R_{CI} - R_{SI})^2 + (G_{CI} - G_{SI})^2 + (B_{CI} - B_{SI})^2} \qquad (13)$$

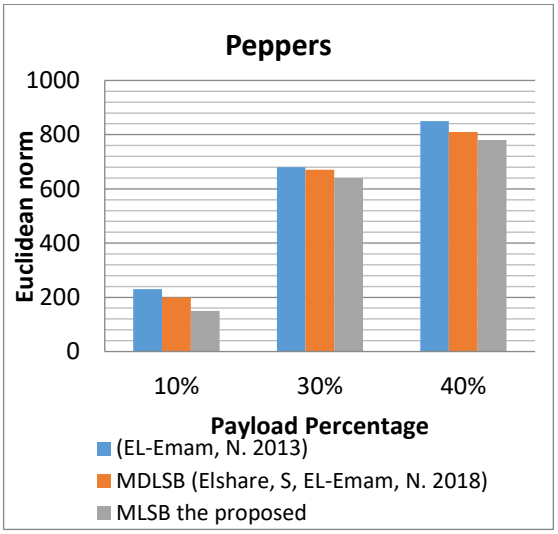
Obviously, the smallest distance has been arrived at when the proposed algorithm is implemented. Moreover, the results show that the maximum difference between the proposed algorithm and the previous work is equal to 70 in the Baboon and Peppers images with a payload percentage equal to 40% and in the Lena image with a payload percentage equal to 30%. However, the minimum difference is equal to 15 between the proposed algorithm and the previous work on the image Peppers with a payload percentage equal to 30%; see Figure 7 (A, B, C).



A



B



C

Figure 7. Evaluation of the effects of the Euclidean norm vs the payload capacity for the proposed algorithm and other research studies.

5.4 Avoiding a WFLogSv attack

The main intention of the suggested hiding algorithm is to conceal a secret message Sm in the color image without arising suspicion that the Stego-Image contains secret information. In this work, the Stego-Image produced by the suggested algorithm IDHA has been examined against the WFLogSv attacker using the "Receiver Operating Characteristic" (ROC) curve Elshare and EL-Emam [25]. The ROC curve is based on two parameters: the probability of false alarms (P_{FA}) and the probability of detections (1-P_{MD}); see Eqs. (14, 15).

$$P_{FA} = \frac{NCI(SI)}{NCI} \qquad (14)$$

$$P_{MD} = \frac{NSI(CI)}{NSI} \qquad (15)$$

where

$P_{FA}, P_{MD} \in [0,1]$, NCI (SI) is the number of Cover-Image s recognized as Stego-Image s, NCI is the total number of Cover-Image s, NSI(CI) is the number of Stego-Image s recognized as Cover-Image s, and NSI is the total number of Stego-Image s.

In the ROC scheme, P_{FA} is presented on the horizontal axis, while (1-P_{MD}) is presented on the vertical axis. A steganography technique is said to be absolutely secure with regard to attackers if the following condition is satisfied:

$$|P_{FA} - 1 + P_{MD}| = \varepsilon, \quad and \quad \varepsilon \to 0$$

The above means that the AUC=0.5, whereas the perfect detection of an attacker is found when $\varepsilon \to 1$; see (Li, B et al., 2011).

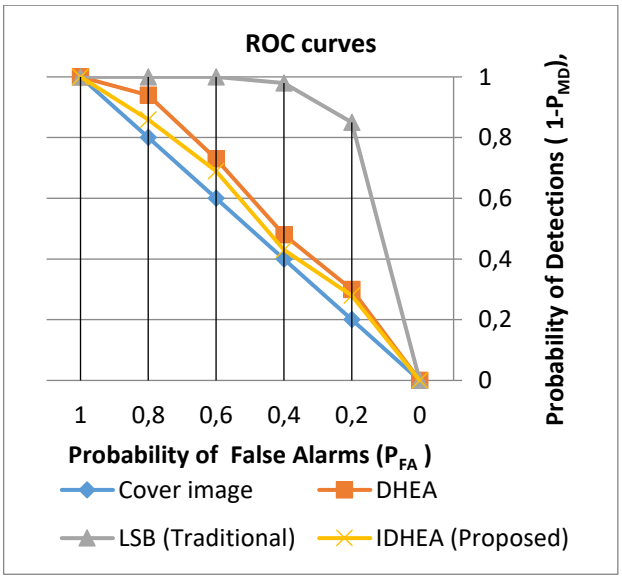


Figure 8. ROC curves of the WFLogSv steganalyser against the proposed hiding algorithm MDLSB and the traditional LSB with a payload of 40% capacity.

Figure 8 shows the performance of the present work with that of other research against the WFLogSv attack using 40% of the payload capacity. It appears that the WFLogSv attack has an excellent detection rate if traditional LSB has been applied, while the WFLogSv attack has poor detection if the present approach has been applied. Additionally, the chance of detecting a secret message using the proposed technique has not exceeded 12%, whereas the chance of detecting the secret message using the DHEA technique (Elshare, S.; EL-Emam, N., 2018) [25] and the traditional LSB algorithm is approximately 15% and 60%, respectively.

5.5 Estimate the detection error (P_E)

The main intention of the suggested hiding algorithm is to conceal a secret message Sm in the color image without arising suspicion that the Stego-Image contains secret information. In this work, the Stego-Image produced by the suggested algorithm IDHA has been examined against the WFLogSv attacker using the "Receiver Operating Characteristic" (ROC) curve Elshare and EL-Emam [25]. The ROC curve is based on two parameters: the probability of false alarms (P_{FA}) and the probability of detections (1-P_{MD}); see Eqs. (14, 15).

The results confirm that the suggested hiding algorithm with IDHA presents high imperceptibility and works against attacks for different payload capacities; see Figure 9. Furthermore, the performance of the present steganography can be achieved by the detection error (PE) expressed in Eq. (13).

The error PE lies in the range [0, 0.5], where zero corresponds to perfect detection and (0.5) to perfect security of the steganographic scheme in Figure 12. The detection error is estimated as a function of the payload capacity based on bits per pixel (bpp) to find the area under the curve (AUC). The results of the suggested algorithm have been analysed and

compared with the previous work (EL-Emam, N. 2015) [7] and (Elshare; S.; EL-Emam, N. 2018) [25].

$$P_E = min\left(\frac{1}{2}(P_{FA} + P_{MD})\right) \quad (13)$$

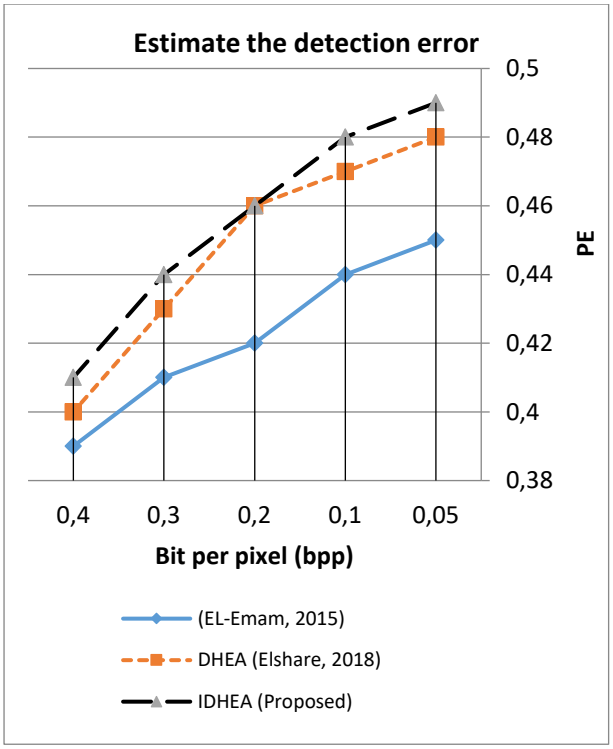


Figure 9. Using P_E as a function of the payload to compare the proposed algorithm and previous work.

The results explain that the average for all bpp of the suggested algorithm IDHEA is equal to 0.456, and it is better than the previous work EL-Emam [7] and Elshare and EL-Emam [25] by approximately 6.8% and 1.6%, respectively. Moreover, the excellent security percentage reaches 98% when bpp=0.05, whereas the worst security percentage reaches 82% when bpp=0.4.

5.6 Avoiding Chi-square (χ2) attack

The main intention of the suggested hiding algorithm is to conceal a secret message Sm in the color image without arising suspicion that the Stego-Image contains secret information.

Westfeld and Pfitzmann [30] suggested a scheme based on a statistical investigation of a pair of values (PoVs) that are exchanged during the hiding process. In this paper, we use this type of inspection to verify how the expected (E_i) and observed (O_i) frequencies of stego pixels are regulated. Chi-square is used for this goal and is measured in the Eq. (14):

$$\chi^2_{k-1} = \sum_{i=0}^{k-1} \frac{(O_i - E_i)^2}{E_i} \quad (14)$$

where, (k) is the number of pairs in the stego image, (k-1) is a degree of freedom and (E_i) is the expected frequency of (i^{th}) pair see Eq. (15).

$$E_i = \frac{1}{2} \underset{c\in\{R,G,B\}}{fr} \{P^c_{2i}, P^c_{2i+1}\}, \forall i = 0, \ldots, k-1 \quad (15)$$

where, $\{P^c_{2i}, P^c_{2i+1}\}$ is the i^{th} pair of the palette colors for pixels $\{P^c_0, P^c_1, \ldots, P^c_{255}\}$. Furthermore, the frequency observes at the (i^{th}) color is shown in Eq. (16)

$$O_i = fr(C_i) \forall i = 0, \ldots, k-1 \quad (16)$$

The probability ($Pr_{\chi^2,k-1}$) bases on Chi-square value (χ^2) with (k-1) degree of freedom is calculated using Eq. (17).

$$Pr_{\chi^2,k-1} = \left(2^{\frac{k-1}{2}} \Gamma\left(\frac{k-1}{2}\right)\right)^{-1} \int_{\chi^2}^{\infty} (t)^{\frac{k-1}{2}-1} e^{-\frac{t}{2}} dt \quad (17)$$

where Gamma ($\Gamma(.)$) is the generalization of the factorial function Eq. (18).

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt \quad (18)$$

It seems that the Chi-square attack on SI with randomly disordered messages produces irregular probability (Pr) values at the beginning and then, as the block size increases, the Pr value eventually drops to zero. Comparisons among two cover images (Baboon and Peppers) and their corresponding Stego-Image s of the previous work EL-Emam and Al-Zubidy [2] and the proposed algorithm are implemented. Furthermore the Pr value eventually drops to zero at the block sizes 20 or 35 or 50 depending on the image color that produces the nearest Pr of hiding the length of Sm around (25% -50%) of the image size for a different kind and message size. Accordingly, steganalysis cannot be detected Sm due to the highly matched of Pr between the SI of the proposed algorithm and CI images see Figures 10 a-10 d.

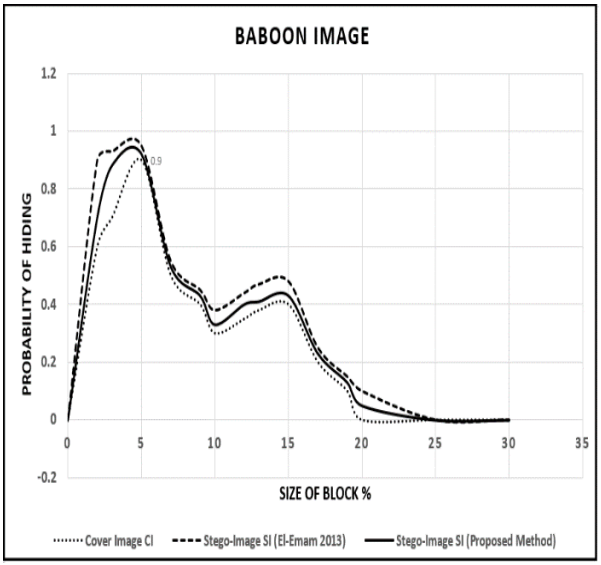


Figure 10 a. Probability of hiding a secret message (Sm) of length 25% of the Baboon image size.

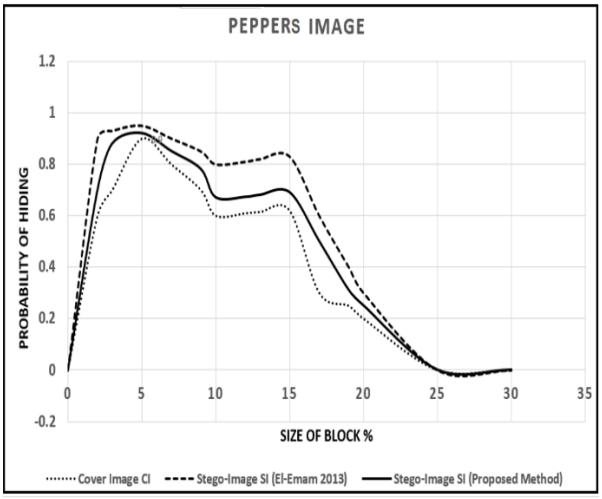


Figure 10 b. Probability of hiding a secret message (Sm) of length 25% of the Peppers image size.

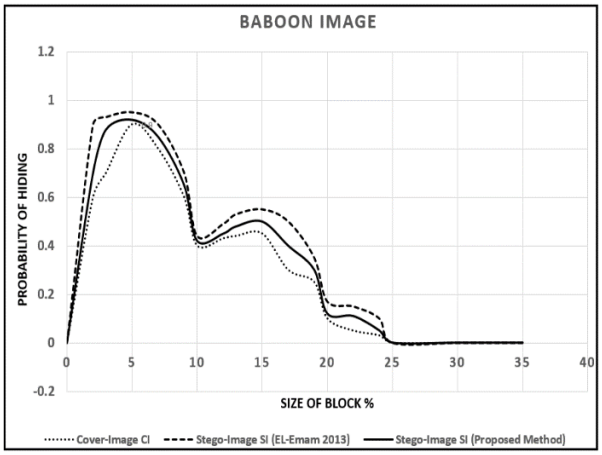


Figure 10 c: Probability of hiding a secret message (Sm) of length 50% of the Baboon image size.

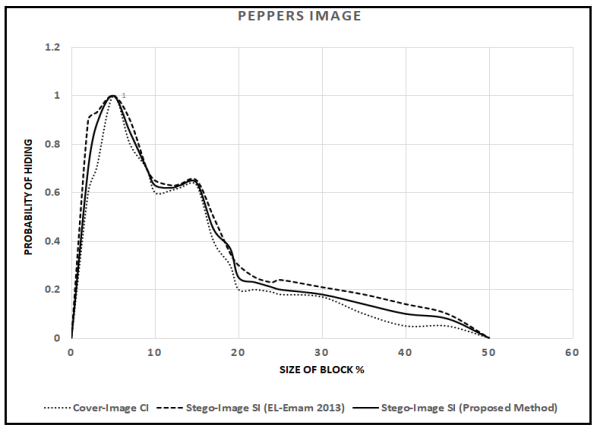


Figure 10 d: Probability of hiding a secret message (Sm) of length 50% of the Peppers image size.

It appears that the difference between the average of the hiding probabilities for both Cover-Image and its corresponding Stego-Image that holds a secret message of length 25% and 50% of the size of image with different block size are around 0.029 and 0.0369 of the Baboon images and 0.048 and 0.033 of the Peppers images respectively. Also, when the length of a secret message is 25% and 50% of the Cover-Image size, then the percentage of the probability of hiding a secret message using the proposed algorithm is less than the same metrics of the previous algorithm EL-Emam and Al-Zubidy [2] is around 2.9% and 3.31% of the Baboon image and 5.6% and 2.78% of Pepper image respectively.

5.7 Calculating the number of levels.

The number of levels is calculated according to Algorithm 1, where the angle (α) of the expansion ration of the cover image size is entered as in Figure 11, and it appeared that the number of levels is increased when the angle (α) is decreased.

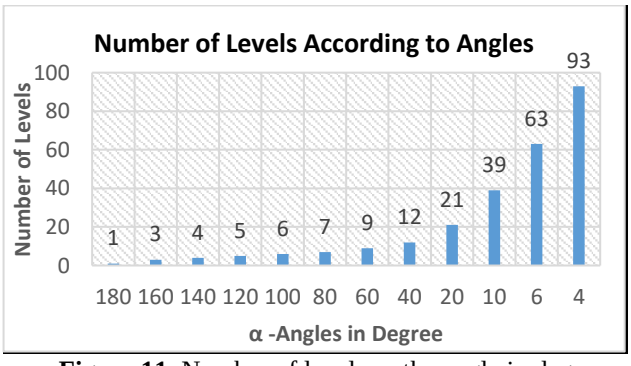


Figure 11. Number of levels vs the angle in degree

Moreover, Table 3 shows the size of a cover image at each level for each angle value. It appears that the enlargement ration of the size of a cover image is small when then angle (α) is small whereas the enlargement ration of the size of a cover image is large when then angle (α) is large.

5.8 General comparisons

In Table 4, various evaluations for many algorithms are presented; it appears that the proposed approach is better than the others for all evaluations.

6. Conclusions

The most important aim of steganography is to maintain the quality of the Stego-Image in such a way that it cannot be distinguished from the Cover-Image. A high-capacity data-hiding scheme has been introduced using the IDHEA algorithm to work under unlimited levels with a gradual or sudden enlargement from one level to the next. Moreover, the proposed technique is performed to confuse attackers under a high-security system, and it divides the secret message Sm over levels relative to the size of the Cover-Image , which is effective in reducing the noise, and the Stego-Image that has been generated after the hiding process cannot be visually distinguished from the original Cover-Image . A large amount of hidden data must increase the number of levels to enable enough space to hide secret data at the next level and hiding is performed randomly with image compression and encryption to make it difficult to detect the secret data. The experimental results show that the proposed scheme has enhanced payload capacity with low complexity and is better than the previous work based on MLS or SLS.

7. Acknowledgement

The authors would like to thank Dr. R. H. Al-Rabeh from Cambridge University for his support and help in this research. This support is gratefully acknowledged.

References

[1] O. Al-Shatanawi, N. El-Emam, “A New Image Steganography Algorithm Based on MLSB Method with Random Pixels Selection,” AIRCC's International Journal of Network Security & Its Applications (IJNSA), Vol. 7, No. 2, pp. 37-53, 2015.

[2] N. El-Emam, R. Al-Zubidy, “New Steganography Algorithm to Conceal a Large Amount of Secret Message Using Hybrid Adaptive Neural Networks with Modified Adaptive Genetic Algorithm,” Elsevier Journal of Systems and Software, Vol. 86, No. 6, pp.1465-1481, 2013.

[3] K. Muhammad, J. Ahmad, N. Rehman, Z. Jan, M. MSajjad, “CISSKA-LSB: Color image steganography using stego key-directed adaptive LSB substitution method,” Springer Journal of Multimedia Tools and Applications, Vol. 76, No. 6, pp.8597–8626, 2016.

[4] A. Darabkh, A. Al-Dhamari, I. Jafar, “A New Steganographic Algorithm Based on Multi Directional PVD and Modified LSB,” KTU Journal of Information, Technology and Control, Vol. 46, No. 1, pp. 16-36, 2017.

[5] N. Sikarwar, “An Integrated Synchronized Protocol for Secure Information Transmission Derived from Multilevel Steganography and Dynamic Cryptography,” International Journal of Computer Science and Telecommunications, Vol. 3, No. 4, pp. 31-36, 2012.

[6] A. Siswanto1, N. Katuk, and K. Ku-Mahamud, “Chaotic-Based Encryption Algorithm using Henon and Logistic Maps for Fingerprint Template Protection,” KUST International Journal of Communication Networks and Information Security, Vol. 12, No. 1, pp. 1-9, 2020.

[7] N. El-Emam, "New Data-Hiding Algorithm Based on Adaptive Neural Networks with Modified Particle Swarm Optimization," Elsevier Journal of Computers & Security, Vol. 55, pp. 21-45, 2015.

[8] N. El-Emam, M. Al-Diabat, "A Novel Algorithm for Color Image Steganography using a New Intelligent Technique Based on Three Phases," Elsevier Journal of Applied Soft Computing, Vol. 37, pp. 830–846, 2015.

[9] N. El-Emam, K. Qaddoum, "Improved Steganographic Security by Applying an Irregular Image Segmentation and Hybrid Adaptive Neural Networks with Modified Ant Colony Optimization," AIRCC's International Journal of Network Security & Its Applications (IJNSA), Vol.7, No.5, pp. 23-47, 2015.

[10] A. Ker, "Steganalysis of LSB Matching in Greyscale Images," IEEE Signal Processing Letters, Vol.12, No. 6, pp. 441-444, 2005.

[11] Y. Latika, Gulati, "A Comparative Study and Literature Review of Image Steganography Techniques," I.J.S.T.E. International Journal of Science Technology & Engineering, Vol.1, No.10, pp. 238-241, 2015.

[12] M. Belhamra, E. Souidi, Mamoun. "Steganography over Redundant Residue Number System Codes," Elsevier Journal of Information Security and Applications, Volume 51, pp.1-10, 2020.

[13] R. Jain, N. Kumar, "Efficient data hiding scheme using lossless data compression and image steganography," AJOL of International Journal of Engineering Science and Technology, Vol. 4, No. 8, pp. 283-241, 2012.

[14] K. Akbay, M. Konyar, S. İlkın, A. Sondaş, "Data hiding using shuffle algorithm and LSB method," 26th Signal Processing and Communications Applications Conference in Turkey, IEEE, pp. 1-4, 2018.

[15] M. Abdulwahed, "Digital image steganography scheme based on SK-LSB substitution and three parameters encryption method," JATIT Journal of Theoretical and Applied Information Technology, Vol. 97, No. 15, pp.4116-4137, 2019.

[16] A. Sahu, G. Swain, "An Improved Data Hiding Technique Using Bit Differencing and LSB Matching," Internetworking Indonesia Journal, Vol.10, No.1, pp.17-21, 2018.

[17] G. Swain, "Very High-Capacity Image Steganography Technique Using Quotient Value Differencing and LSB Substitution," Springer of Arabian Journal for Science and Engineering, Vol. 44, No. 4, pp. 2995-3004, 2019.

[18] S. Nipanikar, V. Deepthi, "Entropy based cost function for wavelet based medical image steganography," International Conference on Intelligent Sustainable Systems (ICISS2017), pp 211-217, 2017.

[19] P. Wang, X. He, Y. Zhang, W. Wen, M. Li, "A robust and secure image sharing scheme with personal identity information embedded," Elsevier Journal of computers & security, Vol. 8, No. 5, pp. 107–121, 2019.

[20] M. Hashim, A. Abdulrazzaq, M. Rahim, M. Taha, H. Khalid, S. Lafta, "Improvement of image steganography scheme based on LSB value with two control random parameters and multi-level encryption," (ICSET2019 Conference, pp.1-14, 2019.

[21] S. Tyagi, P. Anurag, S. Pai, "Multilevel Steganography for Data Protection," Ambient Communications and Computer Systems Sciences," Springer Book, ISBN: 978-981-13-5933-0, 2019.

[22] G. Kasana, K. Singh, S. Bhatia, "Block-Based High-Capacity Multilevel Image Steganography," World Scientific Journal of Circuits, Systems and Computers, Vol. 25, No. 08, 1650091:1-1650091:21, 2016.

[23] P. Xue, H. Liu, J. Hu, R. Hu, "A multi-layer steganographic method based on audio time domain segmented and network steganography," AMCCE 2018 Conference, pp.413-425, 2018.

[24] A. Adeniji, M. Esiefarienrhe, N. Gasale, "Architectural Design of Multi Level Steganography System for Data Transmission," ICCEACT'2014 Conference, pp.78-83, 2014.

[25] S. Elshare, N. EL-Emam,. Modified Multi-Level Steganography to Enhance Data Security. KUST Journal of Communication Networks and Information Security, Vol.10, No. 3, 2018.

[26] K. Bhowal, "Multilevel Steganography to Improve Secret Communication, Digital Image and Video Watermarking and Steganography". IntechOpen Book Chapter 5., 63-76, ISBN: 978-1-78984-168-8, 2019.

[27] R. Castillo, G. Cayabyab, P. Castro, M. Aton, "Block Sight: A Mobile Image Encryption Using Advanced Encryption Standard and Least Significant Bit Algorithm," ICISS'18 Conference, pp.117-121, 2018.

[28] B. Ou, X. Li, Y. Zhao, R. Ni, "Efficient Color Image Reversible Data Hiding Based on Channel-Dependent Payload Partition and Adaptive Embedding," Elsevier Journal of Signal Processing, Vol. 108, pp. 642-657, 2015.

[29] B. Li, J. He, J. Huang, Y. Shi, "A survey on image steganography and steganalysis,". National Kaohsiung University of Applied Sciences Journal of Information Hiding and Multimedia Signal Processing, Vol. 2, No. 2, pp.142-172, 2011.

[30] A. Westfeld, A. Pfitzmann, "Attacks on Steganographic Systems," Springer Lecture Notes in Computer Science Book Series LNCS, Vol. 1768, pp. 61–76, 2000.

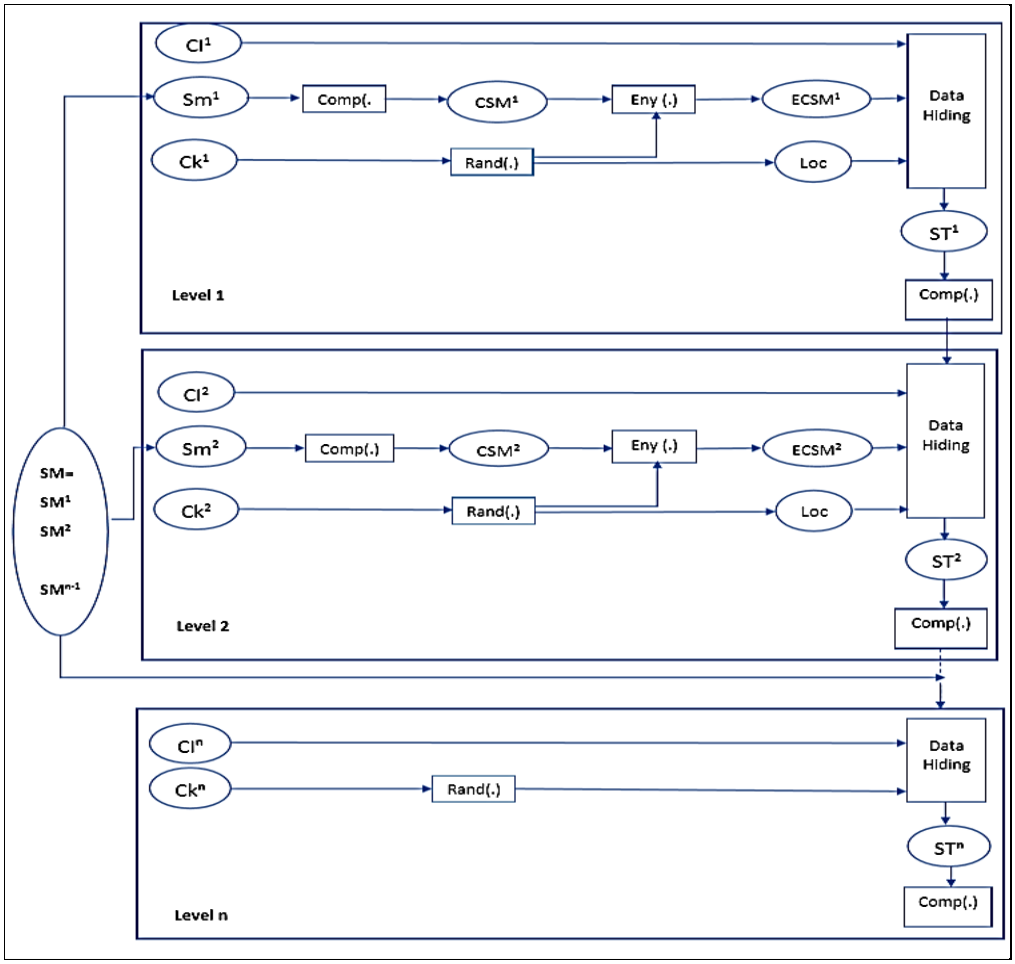


Figure 3. Improved Deep Hiding Algorithm IDHA Architecture.

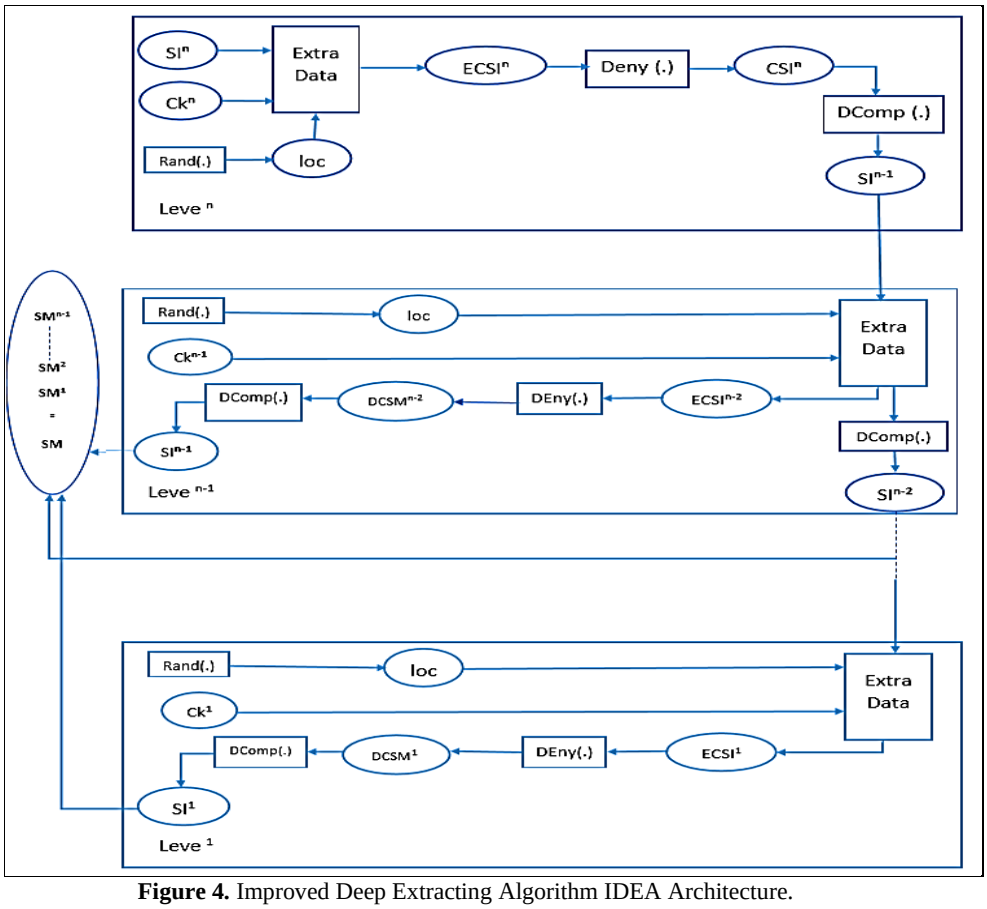


Figure 4. Improved Deep Extracting Algorithm IDEA Architecture.

Table 1. Comparison between the proposed algorithm MLSB and LSB and MDLSB

Image Name	Payload Capacity (bits)	Traditional LSB		MDLSB Elshare and, EL-Emam [25]		The Proposed Algorithm (MLSB)		
		SNR (dB)	PSNR (dB)	SNR (dB)	PSNR (dB)	MSE (dB)	SNR (dB)	PSNR (dB)
People1	3162480	25.3662	27.9431	43.9662	44.1421	0.5874	48.784	50.4414
Tulips	3162480	30.6276	34.9962	48.6376	53.6962	0.0687	55.395	59.760
Lenna	3162480	24.1199	29.2575	44.4199	46.3975	0.1927	50.145	55.282
Onion	3162480	17.2209	24.2278	33.3709	38.4778	1.8527	38.445	45.4528

Table 2. Performance comparison between the proposed algorithm and four embedding algorithms

Images (512×512)	Payload Capacity (bits) $\times 10^4$	PSNR (dB) Ou et al. [28]	PSNR (dB) Li [29]	PSNR (dB) Elshare and EL-Emam [25]	PSNR (dB) of the proposed algorithm MLSB
Lena	2	60.6	59.1	62.7	65.8
	6	55.3	53.8	57.3	60.1
	8	53.8	52.5	55.6	58.3
	12	51.2	50.4	54.8	57.5
	15	49.6	-	53.1	55.7
Baboon	2	56.8	57.1	59.2	63.3
	2.8	54.9	55.2	56.1	57.8
	3.6	53.3	53.3	54.9	56.1
	4.4	51.9	51.9	53.2	55.4
	5.6	49.8	49.9	53.4	53.6
Peppers	2	59.1	57.9	61.2	64.9
	4	55.5	53.7	59.1	59.2
	6	53.1	51.8	55.3	57.5
	8	51.3	50.2	53.7	56.7
	10.5	49.2	-	53.7	54.9

Table 3. Size of cover image at each level according to the change of the angle (α).

#	Angle in degree (α)	Number of levels (images)	Sequence of the size of cover images at each level
1	**10**	**39**	**(2×2), (3×3), (4×4), (5×5), (6×6), (7×7), (8×8), (9×9), (10×10), (11×11) (13×13), (16×16), (19×19), (22×22), (26×26), (31×31), (36×36), (42×42), (50×50), (59×59), (69×69), (81×81), (96×96), (112×112), (132×132),..., (1024×1024).**
2	**20**	**21**	**(2×2), (3×3), (4×4), (6×6), (9×9), (12×12), (16×16), (22×22), (30×30), (41×41), (55×55), (75×75), (101×101), (137×137), (185×185), (251×251), (340×340), (460×460), (622×622), (842×842), (1024×1024).**
3	**50**	**10**	**(2×2), (3×3), (7×7), (14×14), (27×27), (53×53), (104×104), (201×201), (389×398), (753×753), (1024×1024).**
4	**80**	**7**	**(2×2), (5×5), (14×14), (38×38), (103×103), (276×276), (739×739), (1024×1024).**
5	**100**	**6**	**(2×2), (6×6), (22×22), (262×262), (889×889), (1024×1024).**
6	**120**	**5**	**(2×2), (8×8), (39×39), (178×178), (1024×1024).**
7	**140**	**4**	**(2×2), (13×13), (84×84), (550×550), (1024×1024).**
8	**160**	**3**	**(2×2), (24×24), (306×306), (1024×1024).**
9	**180**	**2**	**(2×2), (1024×1024).**

Table 4. Comparison of the proposed algorithm with previous work according to several evaluations

Evaluations	Algorithms			
	IDHEA using MLSB (The Proposed)	**DHEA by Elshare and EL-Emam [25]**	**Traditional MLS**	**ANN AGAUAR by El-Emam and Al-Zubidy [2]**
Payload Capacity	High (If n is large enough, then the length of Sm is also large)	Average	Average	Average
Probability of Extract Sm	1/n (If n is large enough, then the probability of extracting Sm is small)	1/n (If n is large enough, then the probability of extracting Sm is small)	1/n (using a small value of n)	1
Speed of Calculation using a Large Number of Levels	Complexity is increased gradually and time of measure is (T)	Complexity is increased exponentially and time of measure is (2×T)	Complexity is increased exponentially and time of measure is (2×T)	N/A
Working under MLS/SLS	Modified MLS	Modified MLS	MLS	SLS
Calculate Nbpb	Using MLSB (4- neighbours) to find Nbpb	Using MDLSB (9- neighbours) to find Nbpb	Using standard LSB	Using LSB (9-neighbours) to find Nbpb
Using Random or Sequential Data Hiding	Random	Random	Sequential	Random
Check Statistical metrics using MSE (dB)	Low	Average	Average	N/A
Check Statistical metrics using PSNR (dB)	High imperceptible	High (but less than MLSB)	Low	High (Less than MLSB)
Check Statistical metrics using SNR (dB)	High imperceptible	High imperceptible but less than MLSB	Low	High imperceptible (Less than MLSB)
Check Statistical metrics using WFLogSv attack	High imperceptible	High imperceptible but less than MLSB	Low	High imperceptible (Less than MLSB)
Check Statistical metrics using Chi-Square attack	High imperceptible	-	-	High imperceptible (Less than MLSB)
Check Visual attack using Euclidean norm	Low	Low	Low	Low
Dynamic or static process on each level	Dynamic (Using different CK at each level and using a different seed to find random locations)	Dynamic (Using different CK at each level and using a different seed to find random locations)	Static	Static