

NOVEL TECHNIQUE FOR SECURING DATA COMMUNICATION SYSTEMS BY USING CRYPTOGRAPHY AND STEGANOGRAPHY

Walaa H. Al-Qwider¹ and Jamal N. Bani Salameh²

(Received: 17-May-2017, Revised: 20-Jul.-2017, Accepted: 30-Jul.-2017)

ABSTRACT

Information security is becoming more important and attracting much attention nowadays, as the amount of data being exchanged over the internet increased. There are various techniques to secure data communication, but the well-known and widely used techniques are cryptography and steganography. Cryptography changes data into another form that is unreadable by anyone except the intended receiver. Steganography hides the existence of secret data in a cover medium, so that no one can detect the hidden data except the authorized receiver. In this paper, we proposed a new technique for securing data communication systems by combining cryptography and steganography techniques. The cryptography algorithm that was used in this paper is Modified Jamal Encryption Algorithm (MJEa); it is a symmetric (64-bit) block encryption algorithm with (120-bit) key. For steganography, we designed an enhanced form of Least Significant Bit (LSB) algorithm with (128-bit) steg-key. The performance of the proposed technique has been evaluated by considering several experimental tests, such as impressibility test, embedding capacity test and security test. For this purpose, the proposed technique was applied on several 24-bit colored PNG cover images. All experimental results proved the strength of the proposed algorithm in securing the transition of data over unsecure channels to protect it against any attack. Furthermore, the simulation results show the superiority of our proposed algorithm when compared with other algorithms in terms of PSNR and embedding capacity.

KEYWORDS

Cryptography, Steganography, Hybrid system, Information hiding, Information security, MJEa, LSB.

1. INTRODUCTION

Nowadays, the amount of information being exchanged over the network is increasing tremendously and these data are sent mostly over an open channel, which threatens its security [1]. The three main elements that are considered to be the information security requirements are: Confidentiality, Integrity and Availability (CIA). They are widely used as a benchmark for evaluating information systems' security. Information security has gained important attention in recent years to protect the transmitted data from unauthorized accessing or interception. There are a lot of security techniques that are used to ensure the information security requirements; these techniques are cryptography, steganography, watermarking, digital signature and fingerprints. The most well-known and widely used techniques are cryptography and steganography [2]. Cryptography depends on the transfer of secret data from one format to another using certain ways by the sender until it reaches the recipient who works to get it back in its original form, whereas steganography conceals the existence of the secret data in another transmission medium. Cryptography can maintain confidentiality, while steganography can maintain integrity of the transmitted data. Cryptography contains two methods; an encryption method used to convert the original message into an unreadable form with the help of the encryption key and a decryption method used to get back the original message with the help of the decryption key. Cryptographic algorithms can be generally classified into two major groups: symmetric algorithms which use the same key for encryption and decryption and asymmetric algorithms which use different keys [3]. There are a lot of cryptographic algorithms in the literature. [4]-[12] show a review and comparative analysis of various encryption algorithms used for securing information, such as DES, RSA, RC4, RC5, Blowfish, Safer, CAST and Elliptic Curve cryptography.

1. W. H. Al-Qwider is with Computer Engineering Department, Mutah University, Alkarak, Jordan.

2. J. N. Bani Salameh is with Computer Engineering Department, Mutah University, Alkarak, Jordan. Email: jbanisal@mutah.edu.jo, jamalsalameh@yahoo.com.

Steganography is the process of hiding secret or sensitive information inside another carrier in such a way that no one except the authorized user can even detect that there is a secret message inside it [13]. Steganography can use many medium formats as a cover object to hide data in them such as; text, image, video, audio and protocol. The most used cover media are images [14]-[18]. The steganography system that uses digital images as cover files is called image-based steganographic system. Color image and gray image both can be used as cover media, but mostly the color one is used, because it gives more capacity and space for hiding data and its degree of redundancy is high, which makes it very suitable to use in steganography as a cover object [19]. Image steganography techniques can be broadly classified into two categories; namely, spatial domain and frequency or transform domain. In spatial-domain techniques, the pixel values of the cover image are directly manipulated or replaced by the secret message bit. Frequency domain-based methods embed the secret data in the transform coefficients of the cover image. The cover image is first transformed into the frequency domain and then messages are embedded in the transform coefficient [20]. The most common and simplest technique in spatial domain steganography techniques is Least Significant Bit (LSB) technique [21]-[23]. It requires less time and algorithmic complexity compared to other techniques. It takes the least significant bits of each byte on each pixel of the cover image and replaces them with the secret data bits. Number of bytes for each pixel depends on the format used to store the digital image; if 8-bit format is used, then there is just one byte, while if 24-bit format is used, then there are 3 bytes for each pixel. Number of bits taken from each byte to the replacement operation depends on the LSB technique used; some techniques take a fixed number of bits from each byte, whereas other techniques take a variable number of bits [24].

There are a lot of steganography algorithms in the literature; [25]-[26] show a survey and analysis of various steganographic techniques used for information hiding. In the following paragraphs, we will give a brief description about some steganographic algorithms that will be used in the evaluation process to compare them with our proposed algorithm. S. Masud, M. Saifur and M. Ismail proposed a steganography algorithm, where a secret key is used in the selection procedure of the LSB. Also, they used a 24-bit RGB color image as a cover medium. First, the cover image is divided into three matrices (red, green and blue). Also, the secret data is converted into binary representation. The red matrix and the secret key decide where to hide the secret data bits into either green matrix or blue matrix. Each bit of the secret key is XORed with each LSB of red matrix. The resulting XOR value decides that 1 bit of the secret data will be placed with either LSB of the green matrix or the blue matrix. The same process will be continued until the secret data bits are finished. The length of hidden information is stored in the first row of stego-image during the hiding process. In the extraction process, the same secret key must be used. Each bit of the secret key is XORed with each LSB of red matrix of the stego-image. The resulting XOR value decides that 1 bit of hidden information is stored in either LSB of green matrix or blue matrix. The recovery process will be continued depending on the length of hidden information bit stream [27]. Adnan Gutub proposed an RGB image steganography technique based on the pixel indicator. The secret data bits are hidden into two of the RGB pixel channels based on the indication within the third channel. The indicators have been selected in sequence; if the first indicator selection is the red channel in the pixel, the green is channel 1 and the blue is channel 2 (i.e., the sequence is RGB). In the second pixel, if the green channel is selected as the indicator, then red is channel 1 and blue is channel 2 (i.e., the sequence is GRB). In the third pixel, if blue is the indicator, then red is channel 1 and green is channel 2 (i.e., the sequence is BRG) [28]. S. Amritpal and S. Harpal proposed an improved LSB technique for color images by embedding the information into three planes of an RGB image. It replaces bits of the cover image in the order of 2:2:4 of the LSB in three planes (i.e., Red, Green and Blue planes) with the bits of secret data. The ratio 2:2:4 is selected depending on the sensitivity of red and green components of the light being similar and more sensitive as compared to the blue component. Thereby, security is increased and the rate of distortion is lowered in the cover image after hiding the secret message [29].

Cryptography and steganography can be combined together to make a strong security system for securing information from unauthorized access or unwanted intervention, in order to maintain its confidentiality and integrity. There have been lots of works in the literature that combine steganography with cryptography for more security. In the following paragraphs, we will give a brief description of some of them. M. Juneja and P. S. Sandhu designed a robust image steganography technique that is based on LSB insertion and encryption. They first encrypt the user data by using RSA algorithm, then they hide the encrypted data in an image using LSB insertion. The application performs an analysis on the user's library of images. This analysis allows users to hide their data in the image that is less likely

to be vulnerable to stegananalysis [30]. K. Joshi and R. Yadav proposed a new image steganography method in spatial domain combined with cryptography. They use a gray image as a cover image. Firstly, they encrypt the secret message by using Vernam Cipher algorithm, then they hide the encrypted message in a gray image using LSB with Shifting (LSB-S) algorithm. In LSB-S, they use four LSB of the pixel and perform circular left shift operation and XOR operation [31]. Y. Rener, Z. Zhiwei, T. Shun and D. Shilei tried to improve the imperceptibility of the steganography algorithm by combining it with DES encryption algorithm. The secret information is encrypted using DES algorithm, then it is embedded in a gray cover image using LSB steganography algorithm [32]. S. Ush, G. A. Sathish and K. Boopathy proposed a secure triple level encryption method based on combining steganography and cryptography. The data to be sent is first encrypted using the Playfair Cipher method, then the encrypted data will be encrypted again using AES algorithm after that the produced cipher text will be embedded in a color cover image using LSB algorithm and a hiding key to determine where to hide the data [33]. G. S. Charanl, S. S. Kumar, B. Karthikeyan, V. Vaithiyanathan and K. D. Lakshmi proposed a novel LSB image steganography with multi-level encryption. The secret data is first encrypted using Ceaser Cipher technique, then it is encrypted again based on Chaos theory. Thereafter, the encrypted data will be embedded in a color image using 3, 3, 2 LSB replacement algorithm. In this algorithm, each 8 bits of the encrypted data are embedded in one pixel of the cover image, hence the first 3 bits are replaced with 3 LSB bits of red byte, the next 3 bits are replaced with 3 LSB bits of green byte and the last 2 bits are replaced with 2 LSB bits of blue byte [34]. S. A. Laskarand and K. Hemachandran proposed an embedding approach which is a combination of steganography and cryptography. The secret message is first encrypted using transposition cipher method, then the encrypted message is embedded in a cover image using the LSB insertion method. Each 3 bits of the encrypted data will be hidden in one pixel of the cover image; the first bit is replaced with the LSB bit of red byte, the next bit is replaced with the LSB bit of green byte and the last bit is replaced with the LSB bit of blue byte [35]. K. Muhammad, M. Sajjad, I. Mehmood, S. Rho and S. Wook Baik proposed a novel magic LSB substitution method (M-LSB-SM) for RGB image. They convert the image into a hue-saturation-intensity (HSI) color space, then the achromatic component (I-plane) of HIS is divided into four sub-images of equal size. After that, a secret key is used to rotate the four sub-images with different angles. The secret information is divided into four blocks, then each block is encrypted by using multi-level encryption algorithm (MLEA) and embedded into one of the rotated sub-images based on a specific pattern using magic LSB substitution [36].

As mentioned before, using steganography or cryptography alone has some limitation to secure the data communication system. So, combining them together can give a strong security system. Steganography will protect the transmitted data against the suspicion of the attacker by hiding it in a cover image and cryptography will protect the confidentiality of the transmitted data even though the attacker is able to break down the steganography system. The main contribution of this research is developing a new security system that combines steganography with cryptography in order to provide a secure transition of data over open channels.

In this paper, a new technique for securing data communication systems is proposed by combining cryptography and steganography. For steganography, an enhanced form of LSB algorithm was developed to hide the encrypted data in the cover image. The algorithm is based on a secret key used by the sender and the receiver to determine where to hide the encrypted message. So, it is hard for anyone to extract the embedded data without knowing the secret data. The cryptography technique adopted in this research was Modified Jamal Encryption Algorithm (MJEA), which is proposed by Jamal Bani Salameh [37]. MJEA is a novel symmetric block encryption algorithm; it has a block size of 64 bits and 120-bit key. The design of the algorithm is easy to implement and achieves great performance results according to the avalanche effect when compared with other algorithms. The rest of the paper is organized as follows: Section 2 describes the proposed technique in detail. Section 3 shows the experimental results and discusses the efficiency of the proposed mechanism. Finally, section 4 provides some concluding remarks and future work.

2. DESCRIPTION OF THE PROPOSED SYSTEM

In this research, we proposed a hybrid security system that combines cryptography and steganography techniques in order to provide secure transition of data over unsecure channels. In this section, the proposed technique will be introduced in more detail.

Figure 1 shows a block diagram of the proposed system at the transmitter side. As we see in the figure, the hexadecimal representation of the secret message (Sec-msg) is first encrypted by using MJEa under the control of 120-bit key (Enc-Key), where the output is the encrypted message (Enc-msg). As a double security, the Enc-msg is embedded in a cover image (Cov-Img) to get the stego-image (Steg-Img) by using an enhanced form of LSB embedding algorithm and (128-bit) steganography key (Steg-Key). For more security, the Steg-Key is encrypted using MJEa. The encrypted key and the length of the Enc-msg are embedded in the Cov-Img. The output of the embedding process is the Steg-Img that will be sent over the channel to the other side.

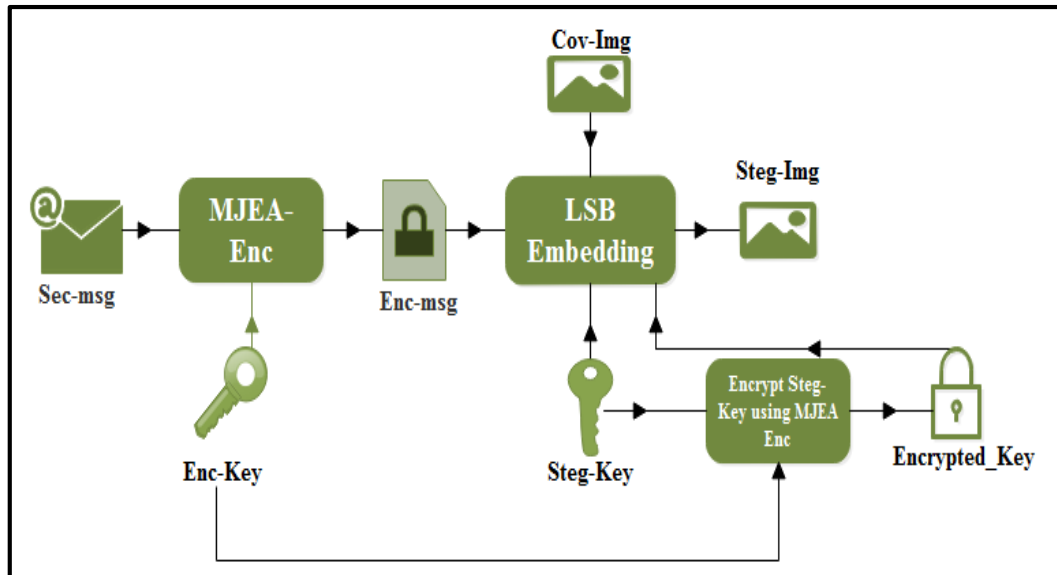


Figure 1. A block diagram of the proposed system at the sender side.

Figure 2 shows a block diagram of the proposed system at the receiver side. The main target for the receiver is to extract the Sec-msg out of the Steg-Img. To do that, the operation is done in a reverse action compared to the transmitter side. The first step in this process is extracting the Enc-Key and the Enc-msg length from the Steg-Img, then using MJEa to decrypt them in order to get the Steg-Key. The LSB extraction algorithm and the Steg-Key are used to extract the Enc-msg from the Steg-Img, then the Enc-msg is decrypted by using MJEa to get back the original Sec-msg that was sent by the sender.

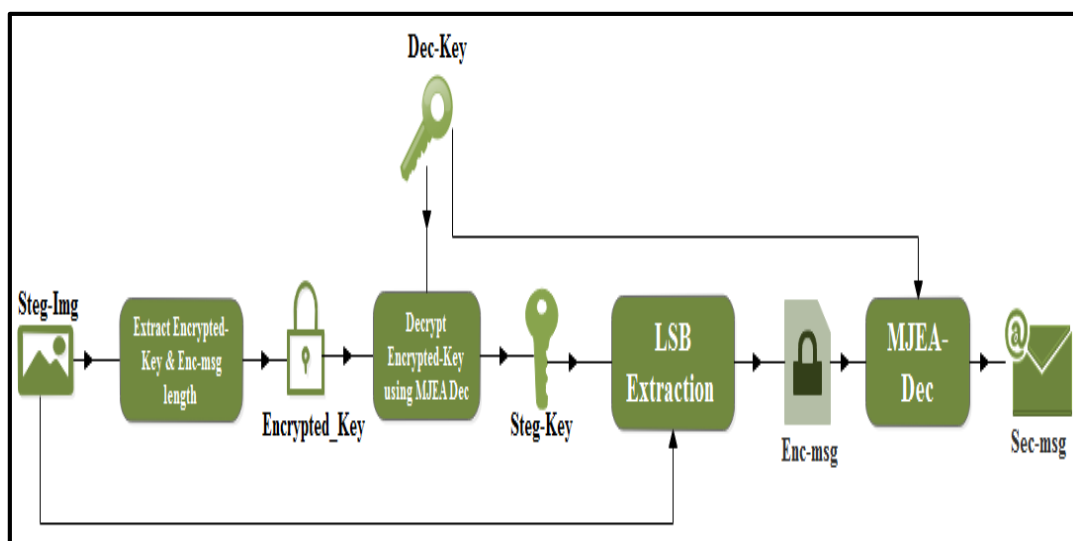


Figure 2. A block diagram of the proposed system at the receiver side.

The proposed system contains four main algorithms: the encryption algorithm, the decryption algorithm, the embedding algorithm and the extraction algorithm. In the following sub-sections, we will give a detailed description of each one of them.

2.1 The Encryption Algorithm

At the sender side, we applied MJEА to encrypt the secret message and the Steg-Key. Figure 3 shows a block diagram of this algorithm. As we see in the figure, MJEА divides the message into (64-bit) blocks, then it encrypts each block separately. All operations in MJEА are XORed on 8-bit words. The 64-bit block of the plain text (Pt) goes in one end of the algorithm, then the algorithm runs to produce the 64-bit cipher text (Ct) at the end. Each (Pt) block is converted into a (Ct) block in 8 rounds under the control of (120-bit) encryption key.

2.2 The Decryption Algorithm

At the receiver side, we applied MJEА algorithm to decrypt the encrypted Steg-Key to get the Steg-Key to be used in the extraction process and to decrypt the encrypted Sec-msg to get the original one that was sent by the transmitter. The decryption algorithm is different from the encryption algorithm in that the S-boxes must be used in the reverse order, as well as the inverse linear transformation and reverse order of the sub-keys. Decryption for MJEА is relatively straightforward beginning with the ciphered text as input which is divided into (64-bit) blocks, then each block is decrypted separately. The (64-bit) block of the Ct goes in one end of the algorithm, and then the algorithm runs to produce the (64-bit) of Pt at the end. Each Ct block is converted into a Pt block in 8-roundes under the control of the same (120-bit) encryption key that was used in the sender side.

2.3 The Embedding Algorithm

This algorithm is used at the sender side to embed the Enc-msg in the least 3-3-2 bits of the Red-Green-Blue components of the Cov-Img under the control of 128-bit Steg-Key. Furthermore, the proposed technique uses the embedding algorithm to hide the encrypted Steg-Key and the message length in the last 17 rows of the Cov-Img array. The full description of the embedding algorithm step by step is shown as a flowchart in Figure 4, which includes the hiding process. As we see in this figure, the inputs for this algorithm are: Enc-msg, Cov-Img, Steg-Key and the Encrypted Steg-Key. The only output out of this algorithm is the Steg-Img that will be sent over the channel to the receiver.

2.4 The Extraction Algorithm

The extraction algorithm takes place at the receiver side. The first step in this algorithm is to extract the (128-bit) Encrypted Steg-Key from the Steg-Img, which will be decrypted by the decryption algorithm to produce the same Steg-Key that was used in the embedding process. This Steg-Key is used in the extraction algorithm to extract the Enc-msg from the Steg-Img. The full description of the extraction algorithm step by step is shown as a flowchart in Figure 5. As we see in this figure, the input for this algorithm is the Steg-Img and the output is the Enc-msg that will be transferred to the decryption algorithm to produce the original Sec-msg that was sent by the transmitter.

3. EXPERIMENTAL RESULTS AND ANALYSIS

The performance of the proposed technique has been evaluated by considering several experimental tests. There are three main metrics that can be used to measure the effectiveness of a given steganography technique: impressibility, embedding capacity and robustness [26].

1. Impressibility: it means that the hidden data cannot be perceived by the human visual system or other statistical means. It is a property in which a person cannot distinguish or detect that the Steg-Img has hidden data.
2. Embedding capacity: it refers to the maximum quantity of secret data that can be hidden inside the cover image without degrading its quality under certain constraints.
3. Robustness: it is the ability of the algorithm to retain the hidden data after many image-related operations, such as cropping, rotating, filtering, compression, etc...

In this research, we used MatLab for simulation, because it supports image processing by using a group of orders under the Image Processing Tool Box.

The evaluation process for the proposed technique will include the following tests:

- Performance analysis of MJEА.

- Visual testing to show how the proposed algorithm works.
- Statistical test, which includes the PSNR (Peak Signal to Noise Ratio) to measure the quality of the image and the MSE (Mean Square Error) to measure the distortion in the image.
- The security test, in which the histogram for all cover images and corresponding Steg-Imgs were calculated then analyzed by comparing them together to see whether the proposed technique is secure against histogram analysis attack.

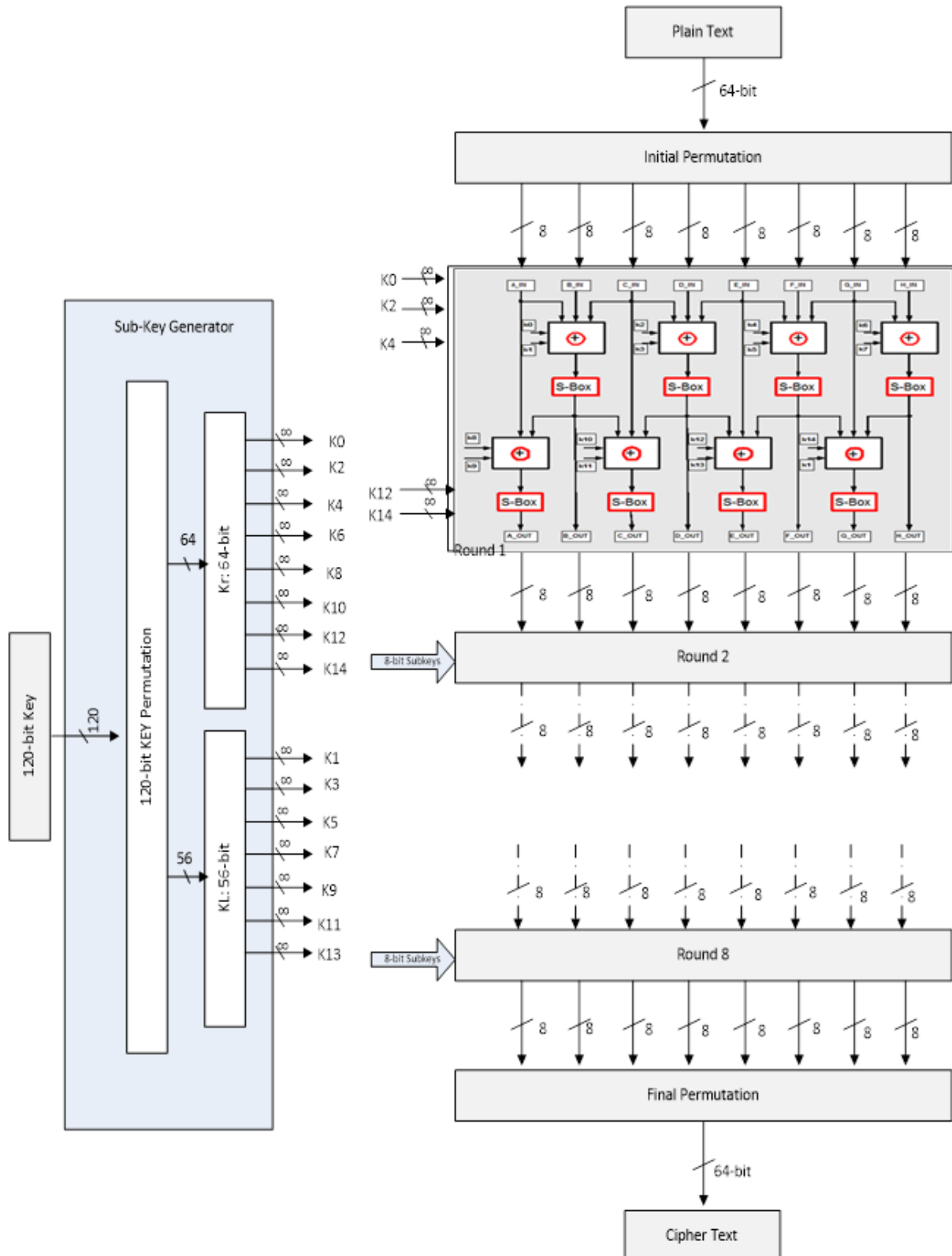


Figure 3. A block diagram of MJEA.

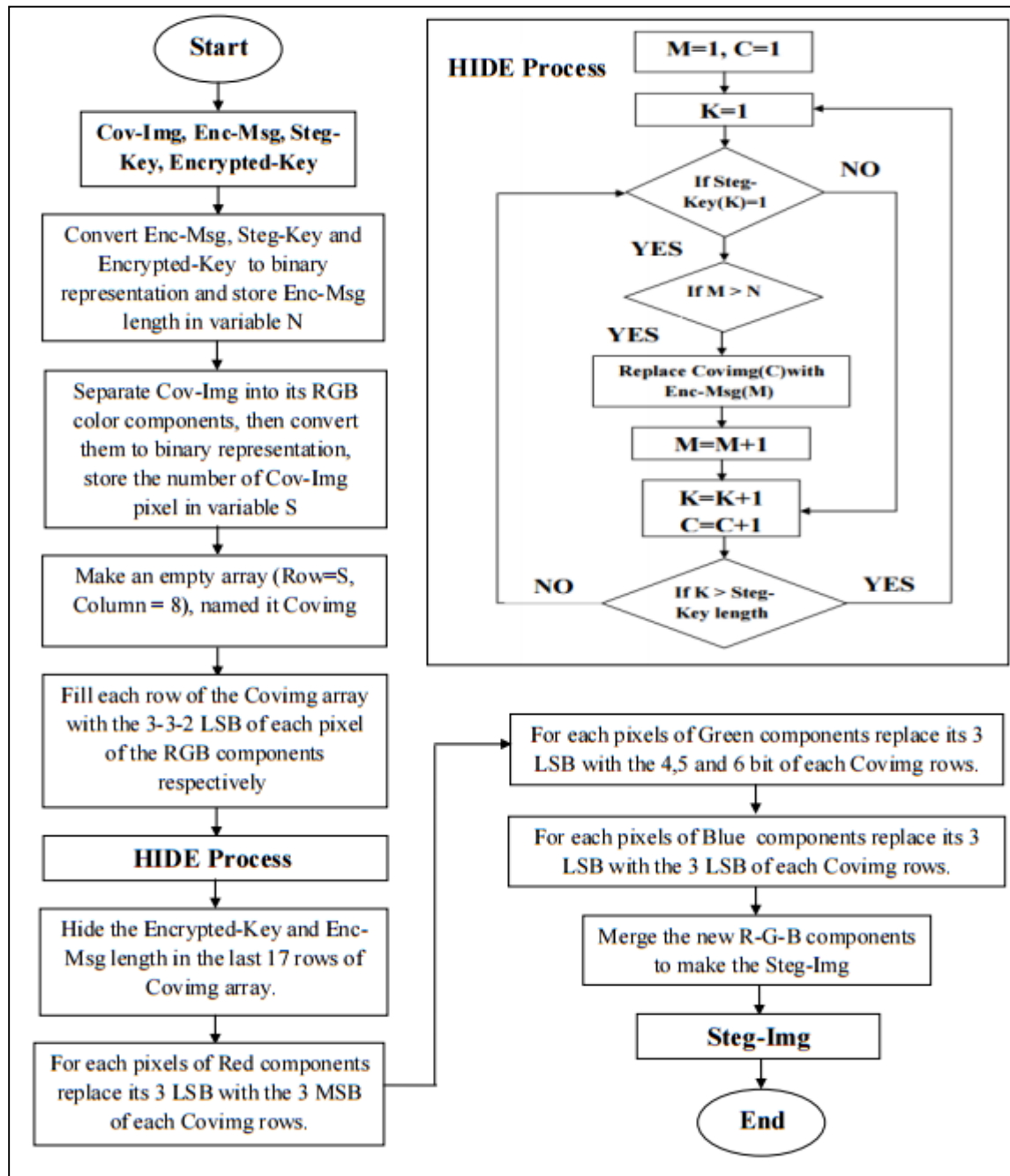


Figure 4. The embedding algorithm.

- Image quality comparison test, in which the proposed algorithm has been compared with other similar LSB algorithms based on PSNR and embedding capacity.

For evaluation purposes, the proposed algorithm was applied on several 24-bit colored PNG (Portable Network Graphic) cover images. This format has been used, because it is a lossless compression technique, which means that the original image will never lose any information when it is compressed during the transmission process over the network. In the following sub-sections, we are going to talk about the simulation results and analysis for all tests conducted.

3.1 Performance Analysis of MJEA

The performance of MJEA was evaluated through a number of experiments in [37]-[38]. MJEA has been analyzed considerably as plain text encryption algorithm through a series of simulation tests. The algorithm thoroughly scrambles the plaintext with the key when run for at least four rounds. MJEA achieved a good Avalanche Effect when tested separately; on average, more than 50% of bits were

changed when we change a bit in the plaintext, key or the ciphertext. A comparison has been conducted between MJEa and different encryption algorithms. Simulation results clearly showed the superiority of MJEa over the other encryption algorithms in terms of Avalanche Effect [37]. Also, MJEa has been analyzed considerably as image encryption algorithm [38]. Experimental results show the possibility of applying MJEa to encrypt digital images. The algorithm was able to achieve high embedding capacity and high quality of encoded image. It was able to replace and transform all pixels in the original-image. On the other side, there was no loss of the image quality after performing the decryption process.

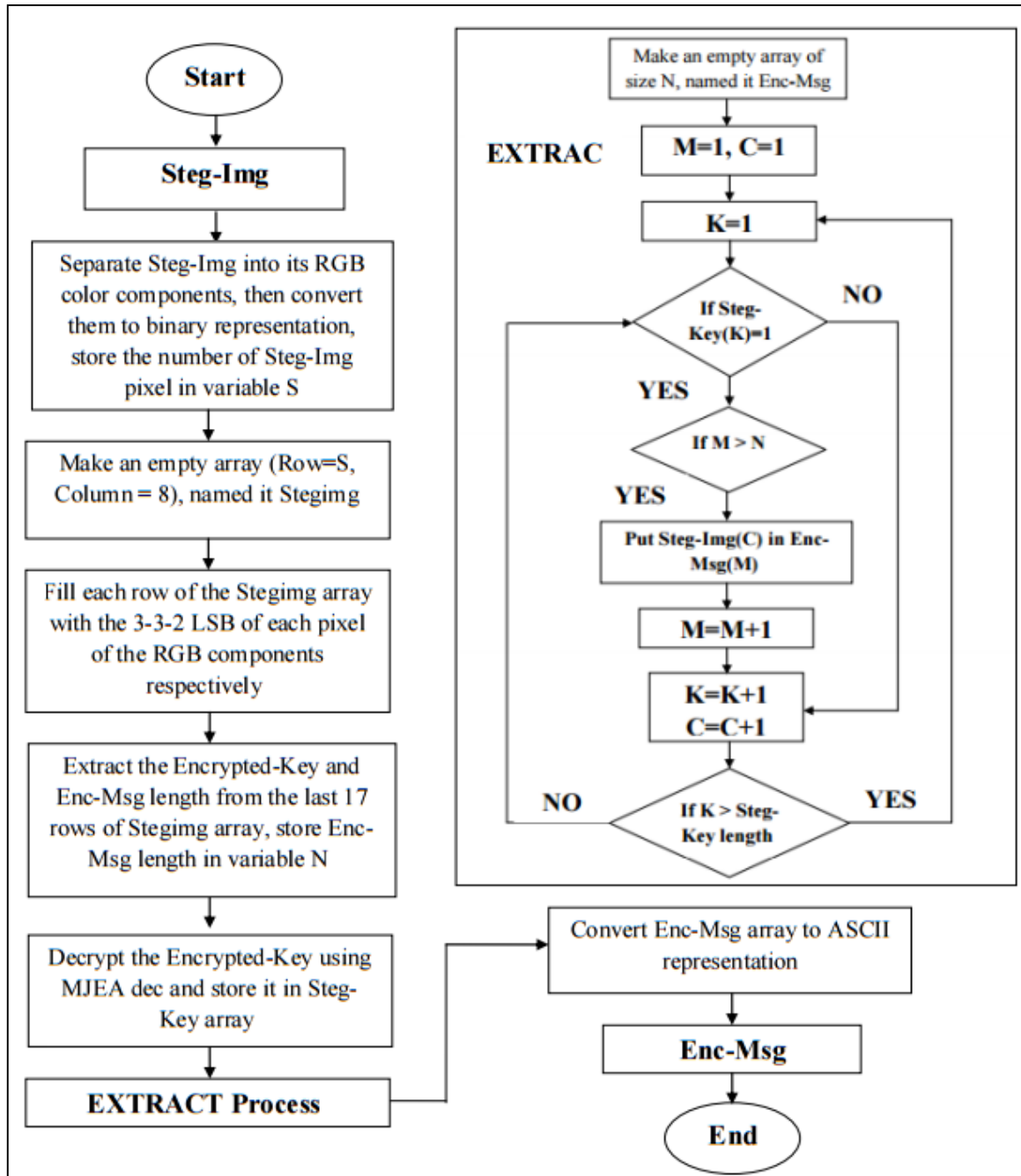


Figure 5. The extraction algorithm.

3.2 Visual Testing for the Proposed Method

In this sub-section, we carried out a series of experiments to show the effectiveness and the correctness of the proposed technique. To do that, we need to prove that the sender is able to hide the secret message in the cover image without being perceived by the human visual system and the receiver is able to extract the secret message from the Steg-Img without any loss. Here, we use three PNG images with different sizes: the first one is Pepper image with (256*256) pixels, the second one is Lena image with (512*512)

pixels and the third one is Animal image with (1024 *1024) pixels. For all experiments, we used the following keys:

The Steg-Key: 88aaa66bbb00ccc77ddd88eee99fff11

The Enc-Key: 111222333444555a666777888999bc

To show the effect of the embedding process on the Cov-Imgs, 2012-byte data representing the Enc-msg is embedded in the three tested Cov-Imgs. The first experiment is done on Pepper image. Figure 6 shows the results at the sender side. Figure 6a shows a sample of the Sec-msg to be encrypted and embedded in a Cov-Img, then transmitted to the receiver. Figure 6b shows a sample of the encrypted Sec-msg. As we see in this figure almost every single bit in the Sec-msg is changed. Figure 6c shows the Cov-Img. Figure 6d shows the Pepper Steg-Img which hides the Enc-msg plus the encrypted Steg-Key. This image will be sent to the receiver side.

4162737472616374 2D496E2074686973 2070617065722C20 616E20696D70726F 76656D656E742069 6E2074686520706C 61696E204C534220 62617365640D0A69 6D61676520737465 67616E6F67726170 6879206973207072	f89b6dfd5a70dd2a c05abe57439b3e5f 6361d471 feecf21 1c35ea227bb556b1 a42d663b7d26f782 2d381820a114dd60 6cc7bb4ff91bb9d9 6adc086d5626605 98f47083dd57fd63 d664e68470562213 2d4d2a6a5b63a89a
a) Sec-msg	b) Enc-msg
	
c) Pepper Cov-Img	d) Pepper Steg-Img

Figure 6. Testing Pepper image at the sender side.

Figure 7 shows the result at the receiver side. Figure 7a shows the Pepper Steg-Img which holds the Enc-msg plus the encrypted Steg-Key that was received from the sender side. Figure 7b shows a sample of the Enc-msg after it was extracted from the Steg-Img by using modified LSB and the help of the Steg-Key. Figure 7c shows a sample of the recovered Sec-msg after it was decrypted by using MJEa and with the help of decryption key. As we see in this figure, the recovered message is exactly the same as the original secret message without any loss.

The second experiment is carried out on Lena image. Figure 8 shows the result at the sender side and Figure 9 shows the result at the receiver side. In this experiment, we repeated the same scenario as we did on Pepper image and found that the receiver was able to recover the Sec-msg that was exactly the same as the original one.

The third experiment is carried out on Animal image. We repeat the same scenario done in the previous two experiments. Figure 10 shows the result at the sender side and Figure 11 shows result at the receiver

side. As we see in those figures, the receiver was able to recover the Sec-msg that was exactly the same as the original one.

	f89b6dfd5a70dd2a c05abe57439b3e5f 6361d471 feecf21 1c35ea227bb556b1 a42d663b7d26f782 2d381820a114dd60 6cc7bb4ff91bb9d9 6adc086d5626605 98f47083dd57fd63 d664e68470562213 2d4d2a6a5b63a89a	4162737472616374 2D496E2074686973 2070617065722C20 616E20696D70726F 76656D656E742069 6E2074686520706C 61696E204C534220 62617365640D0A69 6D61676520737465 67616E6F67726170 6879206973207072
a) Pepper Steg-Img	b) Encrypted Sec-msg	c) Sec-msg

Figure 7. Testing Pepper image at the receiver side.

4162737472616374 2D496E2074686973 2070617065722C20 616E20696D70726F 76656D656E742069 6E2074686520706C 61696E204C534220 62617365640D0A69 6D61676520737465 67616E6F67726170 6879206973207072	f89b6dfd5a70dd2a c05abe57439b3e5f 6361d471 feecf21 1c35ea227bb556b1 a42d663b7d26f782 2d381820a114dd60 6cc7bb4ff91bb9d9 6adc086d5626605 98f47083dd57fd63 d664e68470562213 2d4d2a6a5b63a89a
a) Sec-msg	b) Enc-msg
	
c) Lena Cov-Img	d) Lena Steg-Img

Figure 8. Testing Lena image at the sender side.

Based on the results shown in the figures above, we can note the following:

- At the sender side, the Steg-Imgs and the Cov-Imgs look intact, which means that the hidden data cannot be perceived by the human visual system, which is a good feature of the proposed technique.
- At the receiver side, we were able to extract the Sec-msg from the Steg-Imgs exactly without any loss.

We can conclude from those experiments that the proposed technique works correctly.

	f89b6dfd5a70dd2a c05abe57439b3e5f 6361d471 feecf21 1c35ea227bb556b1 a42d663b7d26f782 2d381820a114dd60 6cc7bb4ff91bb9d9 6adc086d5626605 98f47083dd57fd63 d664e68470562213 2d4d2a6a5b63a89a	4162737472616374 2D496E2074686973 2070617065722C20 616E20696D70726F 76656D656E742069 6E2074686520706C 61696E204C534220 62617365640D0A69 6D61676520737465 67616E6F67726170 6879206973207072
a) Lena Steg-Img	b) Encrypted Sec-msg	a) Sec-msg

Figure 9. Testing Lena image at the receiver side.



4162737472616374 2D496E2074686973 2070617065722C20 616E20696D70726F 76656D656E742069 6E2074686520706C 61696E204C534220 62617365640D0A69 6D61676520737465 67616E6F67726170 6879206973207072	f89b6dfd5a70dd2a c05abe57439b3e5f 6361d471 feecf21 1c35ea227bb556b1 a42d663b7d26f782 2d381820a114dd60 6cc7bb4ff91bb9d9 6adc086d5626605 98f47083dd57fd63 d664e68470562213 2d4d2a6a5b63a89a
a) Sec-msg	b) Enc-msg
	
c) Animal Cov-Img	d) Animal Steg-Img

Figure 10. Testing Animal image at the sender side.

3.3 Statistical Test

The statistical measurements that will be used in this test are PSNR (Peak Signal to Noise Ratio) and MSE (Mean Square Error) measurements. PSNR is a measure of the quality of the image and is measured by comparing the Cov-Img with the Steg-Img. It is calculated by using Equation 1. Higher PSNR value indicates better quality of image (i.e., lower distortion, which decreases the possibility of visual attack by human eyes) [3].

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}_i^2}{\text{MSE}} \right) \text{ (dB)} \quad (1)$$

where MAX_i is the maximum value of the samples which is equal to 255 for a monochrome image having 8 bits per pixel and MSE is the mean square error.

MSE defines as the square of error between the Cov-Img and the Steg-Img and is given in Equation 2. Higher value of MSE means more image distortion.

	f89b6dfd5a70dd2a c05abe57439b3e5f 6361d471 feecf21 1c35ea227bb556b1 a42d663b7d26f782 2d381820a114dd60 6cc7bb4ff91bb9d9 6adc086d5626605 98f47083dd57fd63 d664e68470562213 2d4d2a6a5b63a89a	4162737472616374 2D496E2074686973 2070617065722C20 616E20696D70726F 76656D656E742069 6E2074686520706C 61696E204C534220 62617365640D0A69 6D61676520737465 67616E6F67726170 6879206973207072
a) Animal Steg-Img	b) Encrypted Sec-msg	a) Sec-msg

Figure 11. Testing Animal image at the receiver side.

$$MSE = \frac{1}{M*N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (C(i,j) - S(i,j))^2 \quad (2)$$

where $M*N$ is the image size, $C(i,j)$ is the intensity of the pixel at the Cov-Img and $S(i,j)$ is the intensity of the pixel at the Steg-Img.

To conduct this test, we embedded 2012-byte data representing the Enc-msg in each one of the Cov-Imgs, then we calculated the PSNR and MSE values for all Steg-Imgs produced by the proposed technique with the help of 128-bit Steg-Key. Table 1 shows the results of this test.

Figure 12 shows the results of PSNR and MSE for the three tested images from visualization perspective. The first row shows the Cov-Img's, whereas the second row shows the Steg-Img's after embedding 2012-byte of encrypted data.

Table 1. The PSNR and MSE results.

Images	PSNR	MSE
Pepper (256*256)	57.985	0.14986
Lena(512*512)	64.009	0.03317
Animal(1024*1024)	70.109	0.00820

As seen from the results above, the PSNR values are high and above 50%, which means that the proposed algorithm is strong according to the impressibility test. This result makes it difficult for anyone to notice the existence of hidden data in the Steg-Img. Also, notice that the PSNR value increases as the size of image increases. So, it is recommended to use a bigger size Cov-Img if there is a need to embed a bigger size of Sec-msg. There must be a trade-off between the Cov-Img size and the PSNR value.

3.4 The Embedding Capacity Test

The embedding capacity test calculates the maximum size of the Enc-msg that can be hidden in the Cov-Img and investigates the effect of increasing the size of the Enc-msg on the visual quality of the Steg-Img by calculating its PSNR values. Table 2 shows the maximum embedding capacity for the three cover images (Pepper, Lena and Animal) and Table 3 shows the PSNR and the MSE values for different embedded text sizes that were embedded in the Cov-Imgs.

Figures 13 and 14 show the effect of increasing the embedded text size on the PSNR and MSE values for the Cov-Imgs. Figure 15 shows the effect of increasing the embedded text size on the Steg-Img from a visualization perspective.

As seen from Tables (2-3) and Figures (13-15), the PSNR value decreases as the embedded text size

increases, but the decrease is slow compared to the increase in the text size. When we hide the maximum embedding capacity (32kB text size) in Pepper, we still get a good PSNR result (45dB), still with a good quality image. For Lena and Animal images, the result is still good and accepted and there is an availability to increase the embedded text size up to the maximum size while the quality of the Steg-Img remains good. The MSE results show that Pepper image has the highest value, because we hide the maximum embedding capacity in it. More data embedded in a cover image means more image distortion, which in turn means more MSE values.

		
		
(a) <u>Pepper Img</u> PSNR = 57.985 MSE = 0.14986	(b) <u>Lena Img</u> PSNR = 64.009 MSE = 0.03317	(c) <u>Animal Img</u> PSNR = 70.109 MSE = 0.00820

Figure 12. Results of PSNR and MSE from visualization perspective.

Table 2. Maximum embedding capacity.

Image	Max Embedding Capacity (Byte)
Pepper (256*256)	32768
Lena (512*512)	131072
Animal (1024*1024)	524288

The performance of our algorithm shows a good behavior under the PSNR and MSE tests. The results show that the proposed algorithm has the ability to hide a text size up to the maximum size in the cover images while the quality of the Steg-Img remains accepted without any distortion.

3.5 Security Test

The quality of images could be visually noticed by applying the histogram analysis. In this test, we embedded (2024-byte data) in each one of the Cov-Imgs, then we applied the histogram statistical analysis to get the histograms for all Cov-Imgs and their corresponding Steg-Imgs, whereafter we compared them together.

Table 3. PSNR & MSE values for different text sizes.

Text Size(kB)	Pepper		Lena		Animal	
	PSNR(dB)	MSE	PSNR(dB)	MSE	PSNR(dB)	MSE
1	60.883	0.07543	66.960	0.01667	73.063	0.00405
2	57.985	0.14986	64.009	0.03317	70.109	0.00820
4	54.996	0.29764	61.078	0.06402	67.118	0.01569
8	51.930	0.60484	58.040	0.12784	64.105	0.03171
16	48.989	1.14495	55.048	0.25534	61.061	0.06473
32	45.966	2.21182	52.042	0.50968	58.097	0.12766

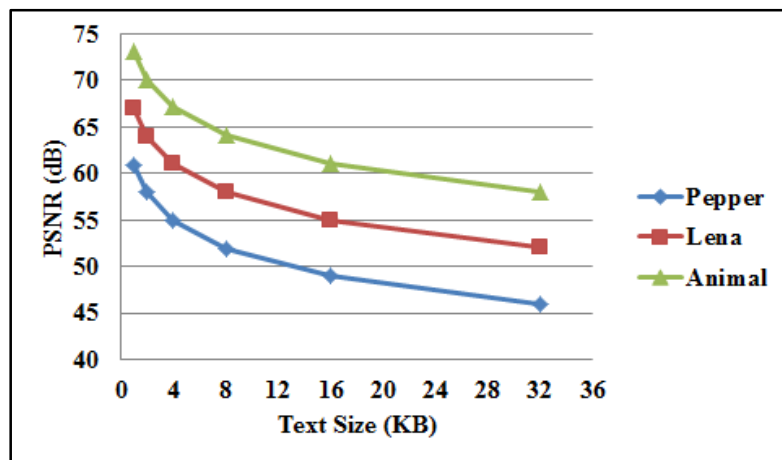


Figure 13. The effect of increasing the embedded text size on the PSNR values.

Figures (16-18) show the histogram analysis for the (Pepper, Lena and Animal) images, where the upper part of each figure represents the histogram analysis for each red, green and blue channel for the Cov-Img and the bottom part represents the histogram analysis for each red, green and blue channel for the Steg-Img. As we can see in all figures, the histograms of the Cov-Imgs look intact with the histograms of the Steg-Imgs. There are no detected visual changes between the original image histograms and the Steg-Img histograms. So, we can say that the proposed technique shows a high degree of security with moderate capacity. This result indicates that it is hard for the stegoanalyst to notice that there is an embedded text in the Steg-Img by analyzing the histograms of the Cov-Img and comparing them with the Steg-Img histograms. This result means that the proposed technique is secured against histogram analysis attack. To add more security level, a secret Steg-Key is used to determine where to hide the secret text in the Cov-Img, so that if the attacker knows that there is an embedded text in the image it will be hard for him to extract this embedded text unless he knows the Steg-Key.

A further security analysis for the proposed method was carried out by using Kirchhoff's principle [39]-

[40]. This test measures the strength of the algorithm against brute force attack.

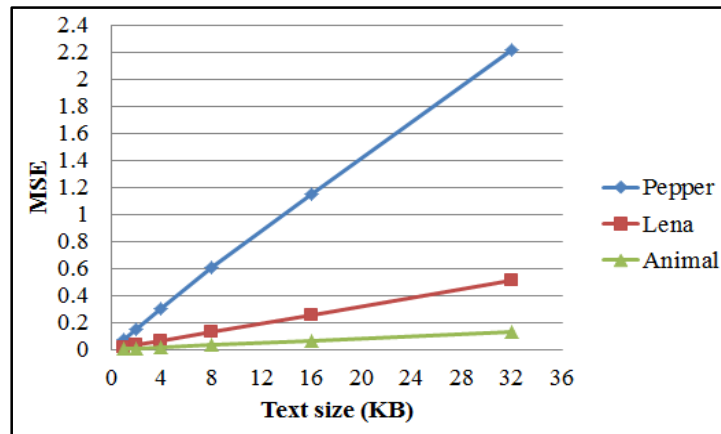


Figure 14. The effect of increasing the embedded text size on the MSE values.










 PSNR= 60.88 MSE= 0.0754	 PSNR=57.985 MSE=0.14986	 PSNR=54.996 MSE= 0.2976	 PSNR=51.930 MSE= 0.6048	 PSNR=48.98 MSE=1.1449	 PSNR=45.966 MSE=2.2118
 PSNR=66.960 MSE=0.0166	 PSNR=64.009 MSE=0.03317	 PSNR=61.078 MSE=0.0640	 PSNR=58.040 MSE=0.1278	 PSNR=55.04M SE=0.2553	 PSNR=52.042 MSE=0.5096
 PSNR=73.063 MSE=0.0040	 PSNR=70.109 MSE=0.00820	 PSNR=67.118 MSE=0.0156	 PSNR=64.105 MSE=0.0317	 PSNR=61.0 MSE=0.0647	 PSNR=58.097 MSE=0.1276
(a) Pepper, Lena and Animal Steg-Img with 1 KB payload	(b) Pepper, Lena and Animal Steg-Img with 2 KB payload	(c) Pepper, Lena and Animal Steg-Img with 4KB payload	(d) Pepper, Lena and Animal Steg-Img with 8KB payload	(e) Pepper, Lena and Animal Steg-Img with 16KB payload	(f) Pepper, Lena and Animal Steg-Img with 32KB payload

Figure 15. The effect of increasing the embedded text size on Steg-Img from visualization perspective.

In this case, the ability of the attacker to break down the system will depend on the length of the used keys. In the proposed system, two keys were used; the Enc-Key (120 bits) and the Steg-Key (128 bits). To calculate the time needed by the attacker to break down the system, we do the following calculation:

- For the cryptography algorithm:

As we know, MJEa is a symmetric cryptography algorithm with (120-bit) key; the attacker needs to produce a 120-bit master key in order to break down this system. This means that the possible number of keys is 2^{120} .

Master key length = 120 bits.

Possible number of keys = 2^{120} .

Assume that the attacker's machine can generate 1 million keys per second.

The time needed to break down the system (years) = $2^{120} / (10^6 * 365 * 86400) = 4.215 * 10^{22}$ (years).

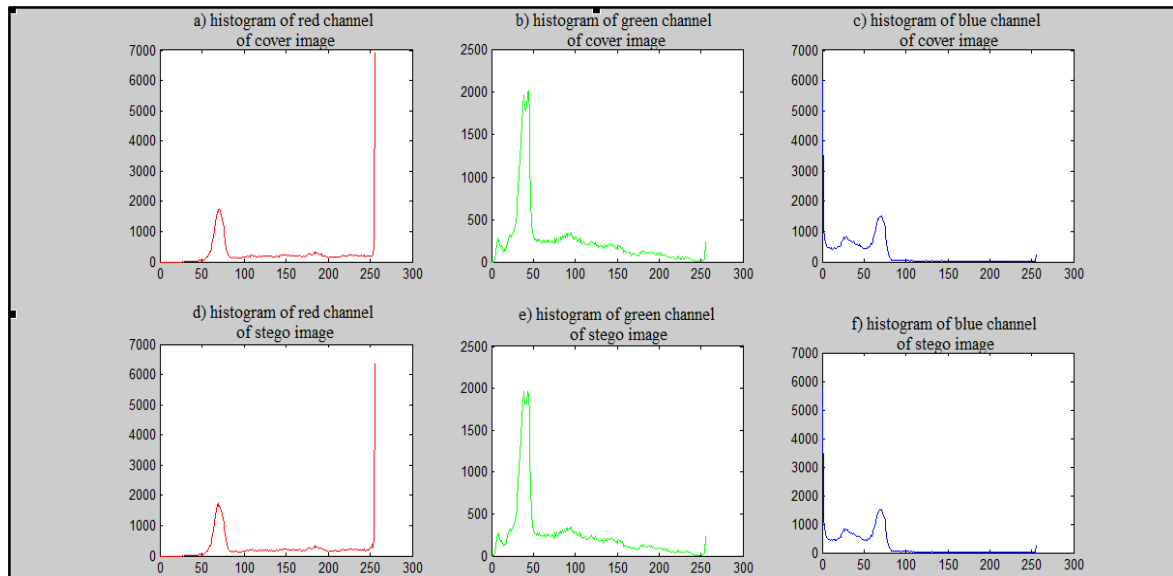


Figure 16. The histogram of the red, green and blue channels of Pepper image.

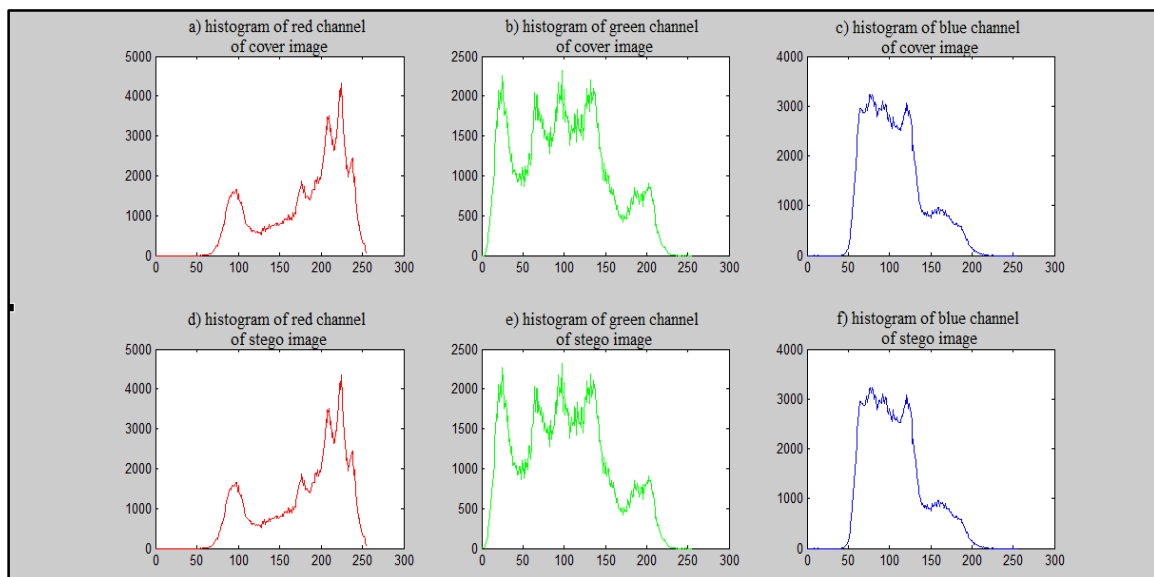


Figure 17. The Histogram of the red, green and blue channels of Lena image.

- For the steganography algorithm:
Our steganography algorithm uses a (128-bit) long steg-key. The attacker needs to produce 128-bit master key in order to break down this system. This means that the possible number of keys is 2^{128} .
Master key length = 128 bits.
Possible number of keys = 2^{128} .
Assume that the attacker's machine can generate 1 million keys per second.
The time needed to break down the system (years) = $2^{128} / (10^6 * 365 * 86400) = 1.079 * 10^{25}$ (years).

This result shows that it takes a very long time from the attacker to break down the system, which indicates that the keys used in the proposed system is strong enough to keep the system secure against brute-force attack.

3.6 Comparison of the Proposed Technique with Other LSB Algorithms

An image quality comparison between the proposed technique and other LSB algorithms similar to our work has been conducted. The nearest works to our proposed algorithm that use LSB steganography technique are: Masud, Saifur and Ismail algorithm, Adnan Gutub algorithm and Amritpal and Harpal

algorithm. Our algorithm will be compared with those algorithms in terms of PSNR and maximum embedded capacity. In order to do this, comparison of the codes for all algorithms is implemented in MatLab. We took the same experimental metrics for all algorithms under the same simulation environment and used the same cover images (Pepper, Lena and Animal) for all algorithms.

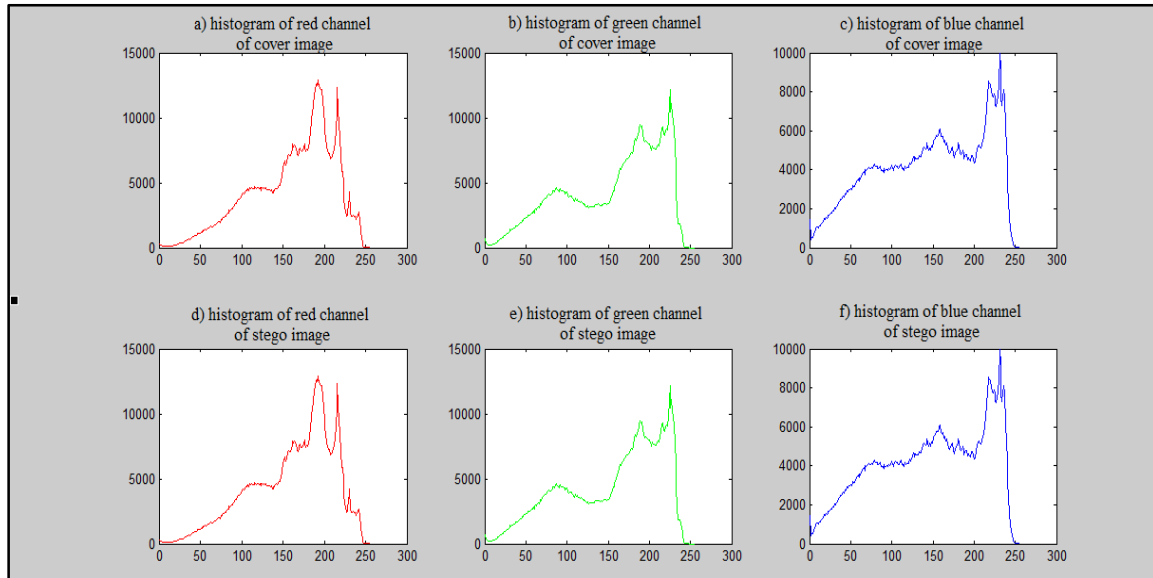


Figure 18. Histogram of the red, green, and blue channels of Animal image.

The first test will be carried out by comparing all algorithms in terms of PSNR. To do this, we embedded (2024-byte) data representing the Enc-msg in each one of the Cov-Imgs. Table 4 shows the PSNR results in dB for all algorithms. The result in the table show that the proposed algorithm has a higher PSNR value with respect to the other algorithms, which means that the quality of the Steg-Img obtained from our algorithm is better than those obtained from the other algorithms. This promising result shows the superiority of our proposed algorithm when compared with other algorithms in terms of PSNR.

The second test will be carried out by comparing all algorithms in terms of payload. To do this we need to calculate the maximum embedded capacity that represents the Enc-msg in each one of the cover images. More embedded capacity in the cover image while still having good visual quality for the Steg-Img means better performance for the algorithm.

Table 5 shows the payload results for all algorithms. It can be noticed from the results shown in the table that the payload of the proposed method is greater than those of the other algorithms except for Amritpal and Harpal algorithm. This result means that this algorithm actually has a greater embedded capacity, but doesn't mean that it is better than our algorithm; it is rather a very simple algorithm with less security because it doesn't have any type of cryptography, the attacker can easily extract the Sec-msg from the Steg-Img.

Table 4. Comparison of the proposed technique with other LSB algorithms using PSNR coefficient in (dB).

Image	Masud,Saifur and Ismail algorithm	Adnan Gutub algorithm	Amritpal and Harpal algorithm	The proposed algorithm
Pepper	59.244	56.361	57.9825	60.055
Lena	64.960	62.431	64.0239	66.252
Animal	70.964	68.103	69.9610	72.311

Table 5. Comparison of the proposed technique with other LSB algorithms using payloads in (byte).

Image	Masud,Saifur and Ismail algorithm	Adnan Gutub algorithm	Amritpal and Harpal algorithm	The proposed algorithm
Pepper	16384	16495	65536	32768
Lena	65536	65903	262144	131072
Animal	262144	262081	1048576	524288

The promising results obtained in this test show the superiority of our proposed technique with respect to the other algorithms.

As a final note on the performance evaluation process, the proposed algorithm has more security than other algorithms because of using the Steg-Key and cryptography. It is hard to extract the Enc-msg out of the Steg-Img without knowing the Steg-Key and if this happened, the message will be unreadable for the attacker, because it is encrypted. The only way to recover the Sec-msg in this case is to break down the security system by getting the encryption algorithm and the Enc-Key.

4. CONCLUSIONS AND FUTURE WORK

In this research, we proposed a novel technique to secure data communication systems by using cryptography and steganography. We used MJEa encryption algorithm to encrypt the secret message before hiding it in the cover image. For steganography, we used an enhanced form of LSB technique to hide the secret message in the cover image.

To evaluate the performance of our proposed technique, we carried out several experimental tests. From the simulation results, we can conclude the following points:

- Based on the visual testing result, the Steg-Imgs and the Cov-Imgs at the sender side look intact, which means that the hidden data cannot be perceived by the human visual system and the receiver was able to extract the Sec-msgs from the Steg-Imgs exactly without any loss, which means that our proposed technique works correctly.
- The histograms of different cover images and their corresponding Steg-Imgs were found to be intact; there was no difference between them. This means that our algorithm is secured against any attack that uses histogram analysis.
- The proposed technique was able to achieve high embedding capacity and high-quality Steg-Imgs.
- The results of the image quality comparison between the proposed technique and other similar LSB algorithms show the superiority of our technique with respect to the other algorithms in terms of PSNR and embedding capacity.
- The encrypted messages are decrypted successfully without any loss, which means that the decoder is efficiently able to recover the original messages.
- The performance of our algorithm shows a good behavior under the PSNR and MSE tests. The results show that the proposed algorithm has the ability to hide a text size up to the maximum size in the cover images while the quality of the Steg-Img remains accepted without any distortion.

Some further research areas could be investigated to improve the performance of the proposed algorithm, such as:

- Using the proposed algorithm with the transform domain embedding technique trying to improve the robustness of the algorithm and to allow using loss compression image format, such as JPEG.
- Using a 32-bit color image instead of a 24-bit one, which will help in increasing the maximum embedding capacity.

- Trying to use audio or video as cover media instead of images.
- Hiding another format of secret data, such as images.
- Evaluating our proposed method under more performance metrics.

REFERENCES

- [1] A. Gutub, "Social Media & Its Impact on E-governance," The 4th Annual Middle East Smart Cities Summit (ME Smart Cities 2015), Dubai, UAE, December 2015.
- [2] N. A. Al-Otaibi and A. A. Gutub, "2-Layer Security System for Hiding Sensitive Text Data on Personal Computers," Lecture Notes on Information Theory, vol. 2, no. 2, pp. 151-157, Engineering and Technology Publishing, June 2014.
- [3] W. Stallings, Cryptography and Network Security: Principles and Practice, Prentice Hall, 2013.
- [4] J. V. Shanta, "Evaluating the Performance of Symmetric Key Algorithms: AES and DES," International Journal of Computational Engineering & Management, vol. 15, no. 4, pp. 43-49, 2012.
- [5] A. Nadeem, "A Performance Comparison of Data Encryption Algorithms," 1st International Conference on Information and Communication Technologies (ICICT), pp. 84-89, 2005.
- [6] N. Alassaf, B. Alkazemi and A. Gutub, "Applicable Light-Weight Cryptography to Secure Medical Data in IoT Systems," Journal of Research in Engineering and Applied Sciences (JREAS), vol. 2, no. 2, pp. 50-58, April 2017.
- [7] A. Gutub and F. A. Khan, "Hybrid Crypto Hardware Utilizing Symmetric-Key & Public-Key Cryptosystems," International Conf. on Advanced Computer Science Applications and Technologies (ACSAT), pp. 116-121, 2012.
- [8] A. Gutub, "Remodeling of Elliptic Curve Cryptography Scalar Multiplication Architecture Using Parallel Jacobian Coordinate System," International Journal of Computer Science and Security (IJCSS), vol. 4, no. 4, pp. 409-425, October 2010.
- [9] A. Gutub, "Efficient Utilization of Scalable Multipliers in Parallel to Compute GF(p) Elliptic Curve Cryptographic Operations," Kuwait Journal of Science & Engineering (KJSE), vol. 34, no. 2, pp. 165-182, December 2007.
- [10] A. A. Gutub, A. Tabakh, A. Al-Qahtani and A. Amin, "Serial vs. Parallel Elliptic Curve Crypto Processor Designs," IADIS International Conference: Applied Computing, pp. 67-74, Fort Worth, Texas, 23-25 October 2013.
- [11] A. A. Gutub, "Preference of Efficient Architectures for GF(p) Elliptic Curve Crypto Operations Using Multiple Parallel Multipliers," International Journal of Security (IJS), vol. 4, no. 4, pp. 46 – 63, 2010.
- [12] A. A. Gutub, "Fast 160-Bits GF(p) Elliptic Curve Crypto Hardware of High-Radix Scalable Multipliers," International Arab Journal of Information Technology (IAJIT), vol. 3, no. 4, October 2006.
- [13] S. Channalli and A. Jadhav, "Steganography an Art of Hiding Data," International Journal of Computer Science and Engineering, vol. 3, pp. 137-141, 2009.
- [14] F. Khan and A. A. Gutub, "Message Concealment Techniques Using Image-based Steganography," 4th IEEE GCC Conference and Exhibition, Gulf International Convention Centre, 2007.
- [15] M. T. Parvez and A. A. Gutub, "Vibrant Color Image Steganography Using Channel Differences and Secret Data Distribution," Kuwait Journal of Science and Engineering (KJSE), vol. 38, no. 1B, pp. 127-142, June 2011.
- [16] W. Abu-Marie, H. Abu-Mansour and A. A. Gutub, "Image-based Steganography Using Truth Table-based and Determinate Array on RGB Indicator," International Journal of Signal and Image Processing (IJSIP), vol. 1, no. 3, pp. 196-204, May 2010.
- [17] N. A. Al-Otaibi and A. A. Gutub, "Flexible Stego-System for Hiding Text in Images of Personal Computers Based on User Security Priority," International Conf. on Advanced Engineering Technologies (AET-2014), pp. 250-256, 2014.
- [18] A. A. Gutub, A. Al-Qahtani and A. Tabakh, "Triple-A: Secure RGB Image Steganography Based on Randomization," 7th ACS/IEEE International Conference on Computer Systems and Applications, pp. 400-403, 2009.

- [19] S. Hemalatha, U. Dinesh Acharya and A. Renuka, "Comparison of Secure and High Capacity Color Image Steganography Techniques in RGB and YCbCr Domains," *International Journal of Advanced Information Technology*, vol. 3, no. 3, 2013.
- [20] M. Kalita and T. Tuithung, "A Comparative Study of Steganography Algorithms of Spatial and Transform Domain," *International Journal of Computer Applications*, no. 1, pp. 9-14, 2016.
- [21] N. Alotaibi, A. A. Gutub and E. Khan, "Stego-System for Hiding Text in Images of Personal Computers," *The 12th Learning and Technology Conference (Wearable Tech/ Wearable Learning)*, Effat University, Jeddah, KSA, April 2015.
- [22] A. A. Gutub, M. Ankeer, M. Abu-Ghalioun and A. K. Alvi, "Pixel Indicator High Capacity Technique for RGB Image-based Steganography," *5th IEEE International Workshop on Signal Processing and Its Applications*, University of Sharjah, U.A.E., 18 – 20 March 2008.
- [23] K. Muhammad, M. Sajjad, I. Mehmood, S. Rho and S. W. Baik, "Image Steganography Using Uncorrelated Color Space and Its Application for Security of Visual Contents in Online Social Networks," *ELSEVIER, Future Generation Computer Systems*, 2016.
- [24] N. Akhtar, P. Johri and S. Khan, "Enhancing the Security and Quality of LSB Based Image Steganography," *5th IEEE International Conference on Computational Intelligence and Computer Networks*, pp. 385-390, September 2013.
- [25] R. Kaur and B. Singh, "Survey and Analysis of Various Steganographic Techniques," *International Journal of Engineering Science & Advanced Technology*, vol. 2, no. 3, pp. 561-566, 2012.
- [26] C. P. Sumathi, T. Santanam and G. Umamaheswari, "A Study of Various Steganographic Techniques Used for Information Hiding," *International Journal of Computer Science & Engineering Survey*, vol. 4, no. 6, pp. 9-25, December 2013.
- [27] S. M. Masud Karim, M. Saifur Rahman and M. Ismail Hossain, "A New Approach for LSB Based Image Steganography Using Secret Key," *14th IEEE International Conference on Computer and Information Technology*, Dhaka, Bangladesh, 22-24 Dec. 2011.
- [28] A. Gutub, "Pixel Indicator Technique for RGB Image Steganography," *Journal of Emerging Technologies in Web Intelligence*, vol. 2, no. 1, Feb. 2010.
- [29] A. Singh and H. Singh, "An Improved LSB Based Steganography Technique for RGB Images," *IEEE International Conference on Electrical, Computer and Communication Technologies*, 5-7 March, Coimbatore, India, 2015.
- [30] M. Juneja and P. S. Sandhu, "Designing of Robust Image Steganography Technique Based on LSB Insertion and Encryption," *International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 302-305, 2009.
- [31] K. Joshi and R. Yadav, "A New LSB-S Image Steganography Method Blend with Cryptography for Secret Communication," *3rd IEEE International Conference on Image Information Processing*, pp. 86-90, 2015.
- [32] Y. Renner, Z. Zhiwei, T. Shun and D. Shilei, "Image Steganography Combined with DES Encryption Pre-processing," *6th International Conference on Measuring Technology and Mechatronics Automation*, pp. 323-326, Wagnaghat, India, 21-24 Dec. 2014.
- [33] S. Ushll, G. A. Sathish Kumal and K. Boopathybagan, "A Secure Triple Level Encryption Method Using Cryptography and Steganography," *International Conference on Computer Science and Network Technology*, pp.1017-1020, 2011.
- [34] G. Sai Charan, N. Kumar, B. Karthikeyan, V. Yanathan and K. Lakshmi, "A Novel LSB Based Image Steganography with Multi-Level Encryption," *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, 19-20 March 2015.
- [35] S. Ahmed Laskar and K. Hemachandran, "High Capacity Data Hiding Using LSB Steganography and Encryption," *International Journal of Database Management Systems*, vol. 4, no. 6, pp. 57-62, 2012.
- [36] K. Muhammad, M. Sajjad, I. Mehmood, S. Rho and S. Wook Baik, "A Novel Magic LSB Substitution Method (M-LSB-SM) Using Multi-Level Encryption and Achromatic Component of an Image," *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 14867-14893, Nov. 2016.
- [37] J. N. Salameh, "A New Symmetric-Key Block Ciphering Algorithm," *Middle-East Journal of Scientific Research*, vol. 12, no. 5, pp. 662-673, 2012.

- [38] J. N. Bani Salameh, "An Investigation of the Use of MJEa in Image Encryption," WSEAS Transactions on Computers, vol. 15, pp. 12-23, January 2016.
- [39] K. Muhammad, J. Ahmad, N. Ur Rehman, Z. Jan and M. Sajjad, "CISSKA-LSB: Color Image Steganography Using Stego-key-directed Adaptive LSB Substitution Method," Springer US, Multimedia Tools and Applications, vol. 76, no. 6, pp. 8597–8626, April 2016.
- [40] K. Muhammad, M. Sajjad and S. W. Baik, "Dual-Level Security Based Cyclic18 Steganographic Method and Its Application for Secure Transmission of Key Frames During Wireless Capsule Endoscopy," Springer US, Journal of Medical Systems, May 2016.

ملخص البحث:

هناك تقنيات مختلفة لتحقيق أمن البيانات، أبرزها وأوسعها انتشاراً التشفير والاختزال. يعمل التشفير على تغيير البيانات إلى شكل آخر تتعذر قراءته إلا من جانب المستقل المقصود. أما الاختزال فيخفي وجود بيانات سرية في وسيط تغطية بحيث لا يمكن لأحد كشف البيانات المخفية سوى المستقل. تم اقتراح تقنية جديدة في هذه الورقة لضمان أمن البيانات في أنظمة الاتصال عبر المزوجة بين التشفير والاختزال. لأغراض التشفير، استخدمنا خوارزمية جمال المعدلة (MJEa). كما قمنا بتصميم نسخة محسنة من خوارزمية للاختزال.

لقد جرى تقييم أداء التقنية المقترحة بإجراء عدة اختبارات تجريبية، مثل: اختبار الحساسية، واختبار سعة التضمين، واختبار الأمان. ولهذه الغاية، تم تطبيق التقنية المقترحة على عدة صور تغطية ملونة. وقد أثبتت النتائج التجريبية جميعها قوة الخوارزمية المقترحة في ضمان انتقال البيانات عبر القنوات غير الأمانة وحمايتها من أي هجمات.

علاوة على ذلك، أظهرت نتائج المحاكاة تفوق الخوارزمية المقترحة على عدد من الخوارزميات الأخرى التي اقترحها باحثون آخرون من حيث نسبة الإشارة إلى الضجيج وسعة التضمين، الأمر الذي يؤكد نجاعة التقنية المقترحة في هذا البحث في نقل البيانات عبر قنوات الاتصال بأمان.