



*Yarmouk University  
Hijawi Faculty for Technology Engineering  
Computer Engineering Department*

**Enhanced Data Hiding Mechanism Based on Randomized Substitution  
over SLSB Technique**

This Thesis Submitted to The Department of Computer Engineering In partial fulfillment of  
the requirements for Master's Degree of Computer Engineering- Industrial Automation

**By**

**Safaa Mohammad Okour**

**Advisors**

**Dr. Mohammad AL-Jarrah, Advisor**

**Dr. Sami Al-Hamdan, Co-advisor**

**January, 2018**

# Enhanced Data Hiding Mechanism Based on Randomized Substitution over SLSB Technique

By

**Safaa Mohammad Okour**

"This Thesis Submitted to The Department of Computer Engineering In partial fulfillment of  
the requirements for Master's Degree of Computer Engineering- Industrial Automation at  
Yarmouk University, Irbid, Jordan"

**Approved by:**

Dr. Mohammad AL-Jarrah (Chairman) .....Chairman  
Associate professor, Computer Engineering, Yarmouk University

Dr. Sami Al-Hamdan .....Member  
Associate professor, Computer Engineering, Yarmouk University

Dr. Abdel-Karim Al-Tamimi .....Member  
Associate professor, Computer Engineering, Yarmouk University

Dr. Ahmed Musa .....Member  
Associate professor, Communications Engineering, Yarmouk University

**January, 2018**

## Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisors Dr. Mohammad Al-Jarrah and Dr. Sami Al-Hamdan for the continuous support of my master study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing this thesis. I could not have imagined having a better advisor and mentor for my master study.

Last but not the least; I would like to thank my family; my parents, my brother, and sisters for raising my moral throughout writing this thesis and my life in general.

© Arabic Digital Library - Yarmouk University

# Table of Contents

<b>Acknowledgements .....</b>	<b>I</b>
<b>Table of Contents .....</b>	<b>II</b>
<b>List of tables.....</b>	<b>III</b>
<b>List of Figures.....</b>	<b>IV</b>
<b>Acronyms .....</b>	<b>V</b>
<b>ABSTRACT.....</b>	<b>VI</b>
<b>CHAPTER ONE .....</b>	<b>1</b>
<b>Introduction.....</b>	<b>1</b>
1.1 Background .....	1
1.2 Steganography.....	2
1.3 Thesis Objectives .....	4
1.4 Contribution .....	4
1.5 Thesis Organization.....	5
<b>CHAPTER TWO .....</b>	<b>6</b>
<b>Background and Literature Review .....</b>	<b>6</b>
2.1 Introduction.....	6
2.2 LSB Algorithm.....	7
2.3 SLSB Algorithm .....	8
2.4 Encryption.....	12
2.5 Testing Criteria .....	13
2.6 Literature review .....	14
2.6.1 Least Significant Bit (LSB) .....	14
2.6.2 Selected Least Significant Bits (SLSB).....	16

<b>CHAPTER THREE .....</b>	<b>19</b>
<b>Approach and Methodology.....</b>	<b>19</b>
3.1 Introduction.....	19
3.2 The Proposed Data Hiding Algorithm.....	19
3.3 Threshold Selection .....	25
<b>CHAPTER FOUR.....</b>	<b>27</b>
<b>Implementation and Discussion.....</b>	<b>27</b>
4.1 Introduction.....	27
4.2 System specifications for the developed tool.....	27
4.3 Software Tools .....	27
4.4 Proposed Algorithm Implementation.....	28
4.6 System evaluation .....	38
<b>CHAPTER FIVE .....</b>	<b>43</b>
<b>Conclusions and future works .....</b>	<b>43</b>
5.1 Conclusion .....	43
5.2 Future work.....	43
<b>References.....</b>	<b>45</b>
المخلص.....	

## List of tables

Table	Page
Table 1: LSB Analysis .....	9
Table 2: SLSB Analysis .....	9
Table 3: Smoothing algorithm example .....	22
Table 4: Dataset image distribution based on size .....	31
Table 5: Sample of images used for testing the proposed algorithm .....	31
Table 6: Threshold sets used to determine the best ranges for algorithm threshold .....	33
Table 7: Average capacity for the test images over different thresholds .....	35
Table 8: Comparison between presented algorithm and SLSB .....	36
Table 9: Proposed model hiding and recovery phases execution time .....	40

© Arabic Digital Library - Yamouk University

## List of Figures

Figure 1: Steganography common implementation methods. ....	3
Figure 2: Block diagram of steganography principles. ....	7
Figure 3: LSB block diagram. ....	10
Figure 4: SLSB block diagram. ....	11
Figure 5: Block diagram for the presented algorithm sequence functions. ....	23
Figure 6: Block diagram for data recovering phase in presented algorithm. ....	24
Figure 7: Developed system user interface. ....	28
Figure 8: Blob detection and pixel randomization process. ....	29
Figure 9: Average PSNR with respect to threshold value. ....	34
Figure 10: Threshold sets sensitivity curve according to their PSNR results. ....	35
Figure 11: Calculated results of MSE for the proposed algorithm and the SLSB one. ....	38
Figure 12: SNR comparison. ....	39
Figure 13: PSNR comparison. ....	39
Figure 14: MSE K_S test Comparison. ....	41
Figure 15: SNR K_S test Comparison. ....	42
Figure 16: PSNR K_S test Comparison. ....	42

## Acronyms

Acronym	Definition
LSB	Least Significant Bit
SLSB	Selected Least Significant Bit
3DES	Triple Data Encryption Standard
PSNR	Peak Signal To Noise Ratio
SNR	Signal To Noise Ratio
MSE	Mean Square Error
SSL	Secure Socket Layer
RGB	Red Green Blue
HSV	Human Visual System

© Arabic Digital Library-Yarmouk University



# ABSTRACT

**Safaa Mohammad Okour, Enhanced Data Hiding Mechanism Based on Randomized Substitution over SLSB technique. MSc. Thesis, Yarmouk University, 2018. (Supervisors: Dr. Mohammad Al-Jarrah, Dr. Sami Al-Hamdan).**

Data transmission has become one of the most demanding areas nowadays, since the growth of the Internet there were more and more demands over the world-wide network. Cloud computing is a major area over the Internet servicing vital fields in social, business, military, science, and others. Focus on data security; especially on transmission protocols used is getting more attention. Many studies have been carried in many areas of data security such as authentication, encryption, data hiding and validation.

In this thesis, we focus on data hiding (steganography) for the purpose of security of transmitted data. As the data hiding increases, the level of security of the hidden data also increases. An improved SLSB technique for hiding data inside images is presented here. This technique uses blob detection point to select a starting point for hiding the data. Then, it scans the covering image in a pseudo-random way to hide data. The randomization function used here utilizes constants derived from the cover image. One color component is chosen at the beginning of the algorithm based on color statistics of the carrier image such that noise induced by hiding is minimized. For more security, the hidden data is subjected to 3DES encryption before merging in the covering image. A pixel that doesn't satisfy certain noise threshold is skipped and a new random pixel location is generated and tested.

Experimental results of the presented algorithm have shown promising enhancement in terms of efficiency and security over existing SLSB algorithm, since we have achieved higher average PSNR (15.901dB) and SNR (18.3633dB) values when compared to the original SLSB algorithm. Our algorithm has also achieved less MSE with an average of 40.8 when compared to SLSB results. Moreover, higher extraction complexity due to the multi different variables used inside the presented algorithm makes our algorithm more secure than the traditional SLSB.

Keywords: Data transformation, Steganography, Data hiding, SLSB, Encryption, Blob Detection.

## Introduction

### 1.1 Background

Data security is one of the most important technology needs. The importance of data security increases every day as a result of a massive spread of Internet services and the huge demands and dependability on the Internet for exchanging data specially the valuable data, which many companies nowadays rely on. Data exchanging like money transfers, bank user account and services authentication, have become one of the most security demanding sectors as many of these services have become via the Internet. The need to protect them from being attacked or even corrupted is a vital interest. Thus, many encryption mechanisms have been developed and combinations of multi techniques have been adopted to reduce the time of encryption and enhance its efficiency while maintaining the complexity of the encryption. As an example, we can mention the Selected Socket Layer (SSL) (Pradeep & Devendra, 2012) authentication and secure tunnels, which are being used by many companies like Cisco (Cisco SSL Appliances, 2017), and Juniper (Cameron & Wyler, 2007).

Information hiding in security research is divided into two basic topics: digital watermarking as Gurpreet and Kamaljeet confirmed (Kaur & Kaur, 2013) and steganography which is the main topic of their research. Atul Kahate considered steganography to be a branch of cryptography (Kahate, 2013), which tries to hide messages within objects, avoiding the perception that there is some kind of structured message.

Combine data hiding (Steganography) and encryption mechanism makes retrieving the data hidden in the object much harder since the data will be unreadable and any attacking techniques will find the results un-expectable and confusing. A lot of researches have been done to enhance the hiding mechanisms, particularly as a message exchange mechanism for top secret data. Some of the newest trends are to use Steganography as the initial phase of opening the secure tunnels instead of predefined certificate or private and public keys, (Shahana, 2013).

In this thesis, we developed an effective and efficient data hiding mechanism that is capable of hiding the data with least noise and maximum complexity. The system is intended to hide text inside an image, with two main features (data randomization, and noise reduction).

## 1.2 Steganography

Steganography came from the integration of the Greek words Stegano, which means sealed, and graphy which means secret writing (Wikipedia, 2017). Steganography is a very old art of binding personal data with another regular data by using some principles and methods.

The main idea of steganography as Ramadhan & Christian presented (Wahaballa *et.al* 2014) is to prevent unauthorized users from noticing and recognizing the hidden data. Later Steganography was adopted as a secret path to send information invisibly. Latika & Gulati explained (Latika & Gulati, 2015) two general ways of steganography as shown in Figure 1; protection against detection and protection against removal. Protection against detection uses some methods to invisibly hide information that does

not affect original data quality. Protection against removal suggests that the method should be capable of resisting popular digital signal processing. This is obvious from the fact that injecting the hidden data will reduce the object's quality and so reduces the image quality.

Steganography techniques are mostly applied to digital file formats, thereby making it popular for concealing information in image and audio files. Steganography can be divided into four main categories; text, image, audio, and protocol. While the first three involves concealing information in the respective file format, the protocol is more advanced and involves using the communication platform or protocol to hide the information.

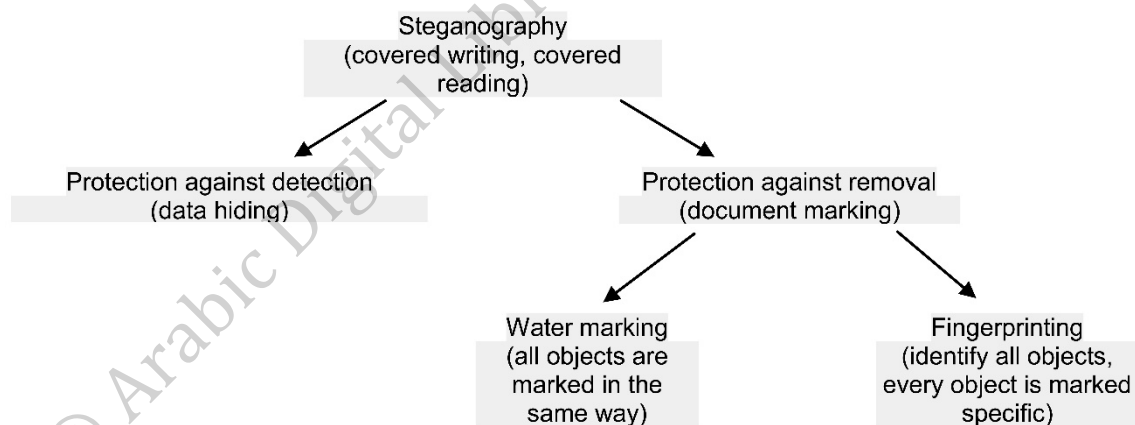


Figure 1: Steganography common implementation methods.

Sumathi and his colleagues classified (Sumathi *et al.* 2013) steganographic techniques and their mode of operations to fall into one of these categories: substitution or replacement, domain transformation, statistical, spread spectrum, and distortion. Just as the name mentioned, substitution techniques involve replacing a certain part of the digital file with the required piece of information that need to be hidden. Transformation domain, on the other hand, encompasses a process whereby the information that

requires concealing is hidden in a frequency space within the file component. Statistical steganographic techniques involve changing the statistical elements of the digital file using various statistical algorithms. Communication using spread spectrum is also another popular steganography technique while distortion involves altering the signal carrying the information and later making a comparison with the original medium content (Sumathi *et al.* 2013).

### 1.3 Thesis Objectives

The main objective of this thesis is to produce a secure algorithm that can protect data without giving any attention to its existence or even its importance. The use of available security encryption algorithm is still vulnerable to many attacks. Thus, we need to hide the existence of data inside some available objects. The objectives of this research can be summarized as follows:

- Produce very fast data hiding mechanism that can be used to transfer data in a secure manner
- Introduce a randomization mechanism with a smart start point location and strong scattering mechanism to Selected Least Significant Bit (SLSB) steganography algorithm. Never to forget the capability to reverse the process.

### 1.4 Contribution

In this thesis, we introduced new steganography approach for hiding data inside an image with a very efficient and very secure way. We enhanced the steganography SLSB algorithm combined with an encryption method to reduce the capability of revealing the data even if it has been extracted. The enhancement to the basic algorithm was in two

major ways: first, reducing the noise caused by the data being injected, and the second, using randomized hiding pattern, which increases the extraction hardness.

## 1.5 Thesis Organization

In this thesis, we present a new data hiding mechanism inside images based on SLSB approach with two major enhancements in the noise reduction and data scattering. The presented approach applies an encryption mechanism to increase the extraction complexity and security level.

The organization of this thesis is as follows: chapter 2 gives a basic background about steganography and literature review for recent published papers. We first explained the main methods used in image steganography. Then we reviewed the data encryption types available. We also highlighted the weakness point of each type. Moreover, the blob detection technique is used in our approach is also described.

Chapter 3 introduces the new approach in details. The tools used to generate the test application are explained in addition to screen shots of the developed application showing the main user interface. Chapter 4 summarizes and explains the obtained experimental results, where we showed the testing system, the images used, and the approved threshold. Finally, chapter 5 gives the final conclusions and directs the reader to suggestions for future work to enhance the presented algorithm performance via reducing the noise and increasing the capacity.

## CHAPTER TWO

### Background and Literature Review

#### 2.1 Introduction

Information distribution and spreading becomes easier and more economic because of the growing of data transferring technology and huge bandwidth of Internet. As a result, people get worried about their privacy and work, but this reveals a critical problem of data security and privacy. Because of that, many mechanisms were developed like encryption and steganography which are techniques that prevents unauthorized users from getting access to private data by hiding the existence of these data. These techniques provide users with methods that can seek and mix their information within other information to make it hard to attackers to recognize (Mstafa & Elleithy, 2015).

Nowadays, many algorithms in data hiding and steganography have been improved in order to protect worthy information. The Human Visual System (HVS) cannot find a slight difference that happens on the cover data including image and video (Xinpeng & Shuozhong , 2005). Unfortunately, many strong analyzing tools in steganography have become available for unauthorized users that are able to retrieve valuable secret information already have been embedded in cover areas. On the other hand, some steganography algorithms have weakness points revealed through steganalysis detectors because of security and embedding efficiency limitation. Figure 2 illustrates the block diagram of steganography principles. Two important factors that every successful and powerful steganography system, should take them into consideration. These two factors are the embedding efficiency and the embedding payload. First, the high embedding

efficiency steganography scheme that has translated to a good visual quality of stego image more than the amount of host (carrier) data are going to be changed. If the viewers notice any clear distortion that will raise the probability of the attacker's suspicion, and some of the stego-analysis tools will detect the secret data easily. These types of schemes are not easy to the steganalysis detectors to detect. The more embedding efficiency scheme, the stronger steganography algorithm achieved. Second, a high embedding payload means large capacity of secret information to be concealed inside cover data. These two factors, embedding efficiency and embedding payload, conflict with each other. When data embedding efficiency goes up, data embedding payload goes down (Mstafa & Elleithy, 2015).

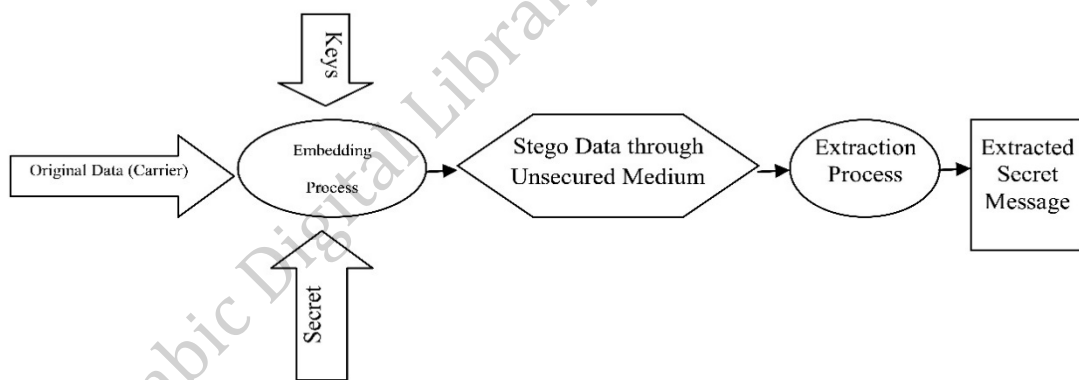


Figure 2: Block diagram of steganography principles.

## 2.2 LSB Algorithm

From its name, Least Significant Bit (LSB) is implemented by replacing the least significant bit with the bit from the intended character Kaza, C (Kaza, C., 2006). It's worth mentioning that LSB algorithm is considered the basic algorithm for the SLSB which was improved based on it. SLSB selects a specific least significant bit from a



pixel's byte, or from a selected section, as proposed by Chang and his colleagues (Chang C. C *et al.* 2009). We have described LSB algorithm in Example 1 below, through hidden character (binary value of 111) in image into 3 pixels from type 24 bitmap. The process of LSB is shown in Figure 3. These operations show the style of LSB; which are made through converting the text message into binary stream and put all bits from the text in LSB (least significant bit) in each pixel, as Siper and his colleagues proposed (Siper *et al.* 2005).

Example 1:

Original pixels:

00100111 11101001 11001000 - pixel.

00100111 11001000 11101000- pixel2.

11001000 00100111 11101000 - pixel3.

Pixels after using LSB:

00100110 11101001 11001001– pixe1.

00100110 11001000 11101001– pixe2.

11001000 00100111 11101001– pixe3.

### 2.3 SLSB Algorithm

This algorithm is a modified and improved version of LSB algorithm, through hiding one bit in the least significant bit of each selected (let's say the green) color of a pixel. Ni and his colleagues showed that this algorithm gives a stego-image with a high visual quality, high hiding capacity (Ni *et al.* 2008). Ibrahim and his colleagues (Ibrahim *et al.* 2009) indicated that changes three color component of each pixel, causing a distortion in the image. Although this distortion is not clear for the human eye, it can be detected by analysis and histogram techniques. Suppose that we have a cover image pixel

(A8A8A8), with a binary value 10101000-10101000-10101000, and we use least bit with to hide one data bit in each color component. Let us assume that we will hide the data (111). According to LSB, the resulting pixel after hiding data (10101001-10101001-10101001) is shown in Table 1. And the resulting pixel after hiding the data according to SLSB is shown in Table 2.

Table 1: LSB Analysis

	Hexadecimal	Decimal	Red	Green	Blue
Original pix	A8A8A8	11053224	168	168	168
Modified pix	A9A9A9	11119017	169	169	169

Table 2: SLSB Analysis

	Hexadecimal	Decimal	Red	Green	Blue
Original pix	A8A8A8	11053224	168	168	168
Modified pix	A8AFA8	11055016	168	175	168

According to Roque and his colleagues (Roque *et al.* 2010), changing the three least significant bits in the pixel, which cause a small distortion, put a leap in the difference between two colors. SLSB method is more effective in using only one color component in a pixel. Using the same example, the resulting pixel after hiding data is (10101000-10101111-10101000) as shown in Table 2. Thus, the image less distorted because one color component of the pixel has been changed. Ker concludes that data detection is very hard (Ker. 2005). The process of SLSB is shown in Figure 4.

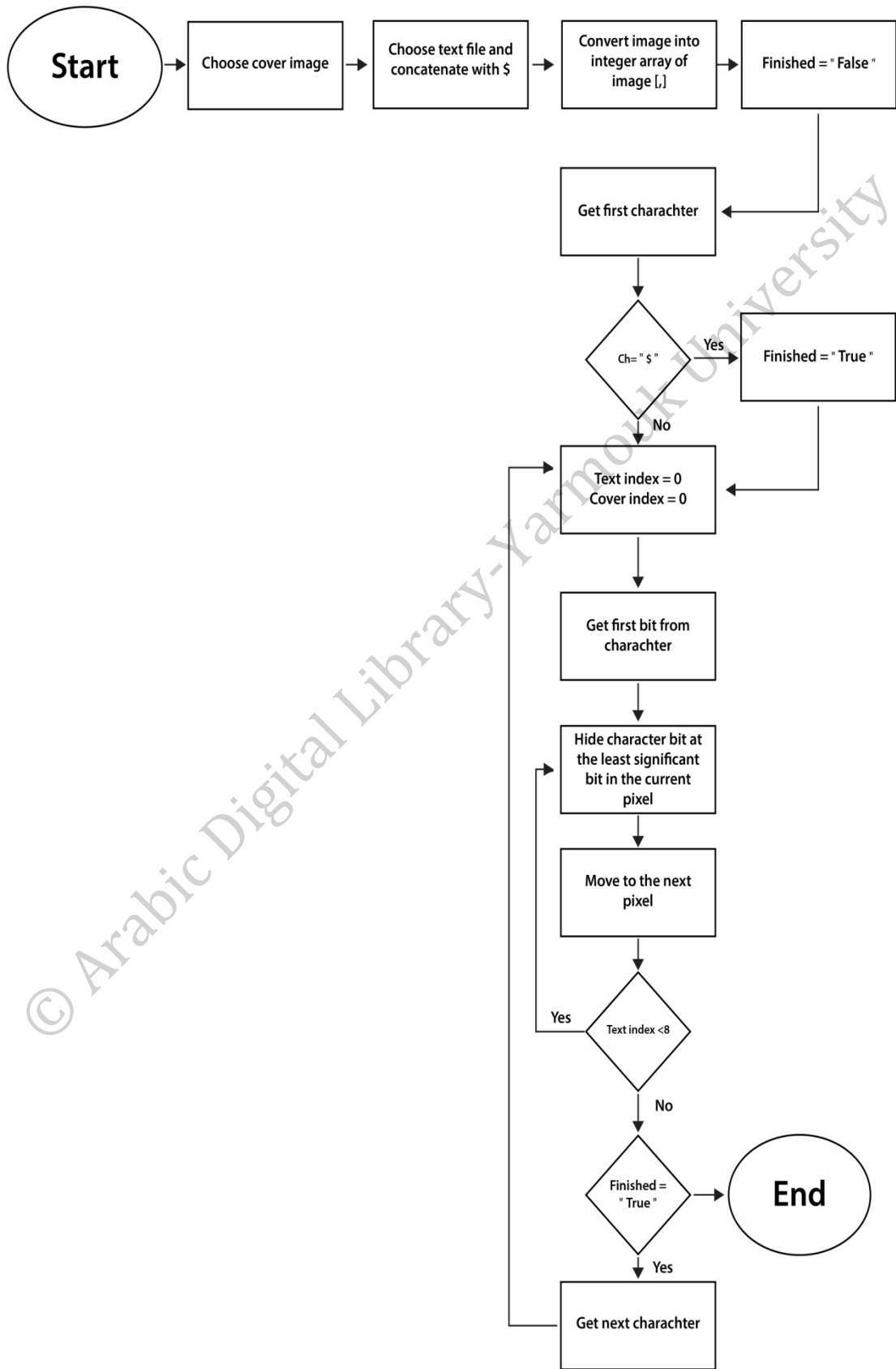


Figure 3: LSB block diagram.

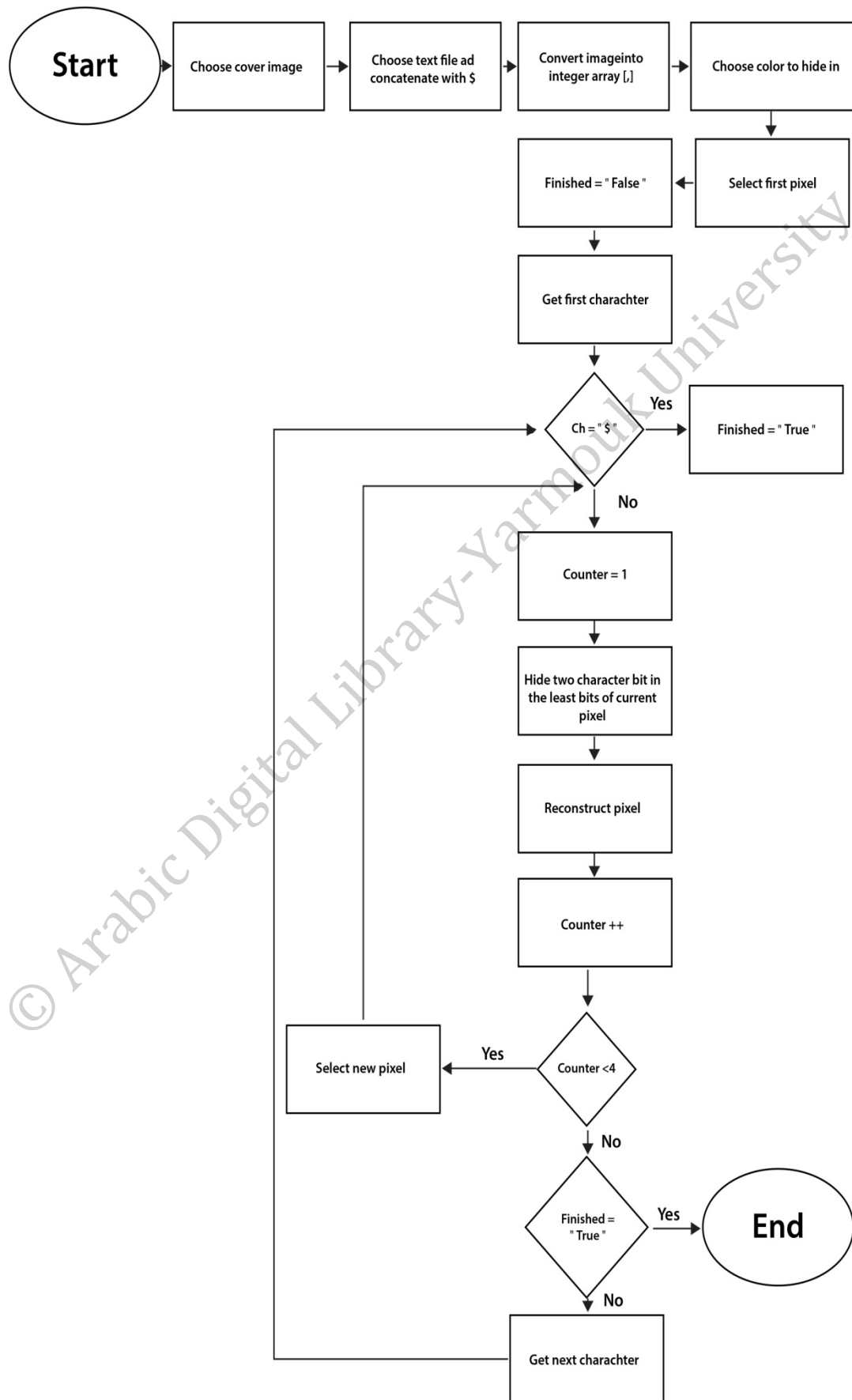


Figure 4: SLSB block diagram.

## 2.4 Encryption

Simmons (Simmons, 1979) clarified that symmetric encryption is the older and the most famous technique. Encryption utilizes a secret key which can be a number or a text or any random combination of both, that is applied on important data messages to convert its content into unreadable data (Handschuh *et al.* 2007). This method of encryption is simple, as it depends on the same secret key for encrypting and decrypting the data. Symmetric encryption can be classified into two types including sequential encryption and block encryption. The sequential encryption converts each symbol in the data into another value. Therefore, it can be vulnerable to brute force attack. The second type is the block encryption which is considered the most used way as it overcomes the standard attacks by encrypting each block of data in a different sub key. Therefore, it is very hard for breaking the secret key.

Poweski and his colleagues (Poweski *et al.* 2009) had a problem that, one cannot use the secret key via the Internet as it may fall in the wrong hands, since anyone can take this key and decrypt the secret message, and to resolve this problem introduced asymmetric encryption technique which has two different keys, one for encryption and the other for decryption.

Asymmetric encryption uses a public key and a private key. The receiver knows the private key. But the public key can be revealed by others as Bellare and his colleagues (Bellare *et al.* 1995) explained. Let's say an encrypted message is sent; only the receiver with the private key is capable of decrypting and reading it correctly, as implemented by Inzunza-González and his colleagues (Inzunza-González *et al.* 2009).

In this thesis, we have to choose an encryption mechanism with high security level but in the same time maintain simplicity. Therefore, we used the symmetric encryption (3DES) because it met our objectives. 3DES considered to be a very strong encryption mechanism but without the complexity of sharing multi keys as asymmetric encryption especially when they demanded sending the needed parameter for extracting the data on a secure tunnel.

## 2.5 Testing Criteria

To measure the performance of the proposed algorithm, the following testing metrics will be utilized:

- Histogram analysis: A histogram shows the distribution of pixels color components inside the image over the range of its values. Using this analysis, we will determine the overall change variation between the cover image color component distribution and the stego-image. As each color component varies from 0 to 255, we split the pixels according to their color components into 25 groups, each group contains 10 color values (0-9, 10-19, 20-29, 30-39, ... 240-249, 250-255). Then we subtract the histogram of the stego image from the histogram of the original image. This will clarify the change effect of color distribution in the stego image over the original one.
- Mean Square Error (MSE): This metric measures the average of the squares of the errors, that is, the difference between the estimator and what is estimated (the stego-image and the cover image). MSE can be calculated as follows:  
Let  $m \times n$  be the width and the height in pixels (integers) of image  $I$  of 24-bit, and its noisy approximation  $K$ , MSE is defined as:

$$MSE = \frac{1}{m * n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \quad (1)$$

- Peak signal to noise ratio (PSNR): This metric describes the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. This mechanism usually is used to determine the quality of reconstructing the signals, like video coding and data encoding. In our case, we used it to evaluate the quality of the stego-image compared to the original one. The PSNR calculation depends on the MSE metric and can be defined as follows:

Let  $MAX_i$  be the maximum possible pixel value of the image, then PSNR equals:

$$PSNR = 10 \cdot \text{Log}_{10} \left( \frac{MAX_i^2}{MSE} \right) = 20 \cdot \text{Log}_{10}(MAX_i) - 10 \cdot \text{Log}_{10}(MSE) \quad (2)$$

## 2.6 Literature review

This section discusses previous work on data hiding using image or object for hiding and their different approaches.

### 2.6.1 Least Significant Bit (LSB)

LSB is among the existing substitution steganography techniques (Jain & Kumar, 2012). Jain AND Kumar applied LSB for data hiding on gray image. Using color images raises data capacity

Champakamala and his colleagues explained (Champakamala *et al.* 2012) that LSB achieves its objective by altering the coloring scheme of an image, thereby making it

difficult for the human eye to discern the content. They found that LSB technique does not distort the image comparing to the Most Significant Bit (MSB) techniques, for that they proposed a new LSB technique with very low distortion factor. Priya and his colleagues proposed (Priya *et al.* 2012) a method of further eliminating distortion in LSB whereby two pixels are used for embedding data. This method combines the operation of one pixel and a function value of the other pixels in grayscale images.

Another LSB technique proposed by Ker (Ker, 2005) involves a matching method that is integrated with the steganalysis process. This method is precipitated by the notion that it is sometimes possible to detect the concealed information using statistical analysis, such as the use of Chi-Squares as well as Regular and Singular. In his approach, Ker suggested that the effectiveness of the matching technique is enhanced using histogram characteristic function (HCF) to detect patterns in grayscale images.

Yang and his colleagues (Yang *et al.* 2008) described another LSB technique known as adaptive data hiding whereby it works on the principle of hiding information within the edges of an image. The technique works by calculating two-pixel values; one from the smooth edge (where color changes between neighbors in fixed manner) and the other from the sharp edge (where color changes between neighbors in great and noticeable manner) of the image. The difference in the pixel value is used to determine the pixel that would be substituted with the information to be concealed. There is other LSB technique and algorithms that utilize the RGB coloring scheme. In particular, Babita & Kaur proposed (Babita & Kaur, 2009) proposed a method that uses the median of the respective code values of the four-color schemes to determine the hiding criteria of the cover image. They described another LSB algorithm that utilizes the texture format of an image. In this case, the simple texture generates 3 LSB channels while the complex texture generates 4 LSB channels.



Sumathi and his colleagues (Sumathi *et al.* 2013) used Pseudo Random Number Generator (PRNG) to embed data in image whereby a red pane of the image is used to conceal the intended information. Medeni discussed another technique (Medeni *et al.* 2010) known as Pixel Value Differencing (PVD), which works by combining various blocks that connect data pixels for embedding. Thiyagarajan (Thiyagarajan *et al.* 2013) makes use of 3D models to triangulate and embed data in the newly triangle mesh using a distinct transformation algorithm.

### 2.6.2 Selected Least Significant Bits (SLSB)

SLSB is regarded as a spatial domain algorithm that serves to improve the performance and the efficiency of the LSB. As a spatial domain filtering mechanism, SLSB is quite simple and fast to implement. According to Warade and his colleagues (Warade *et al.* 2014) LSB has one deficiency in that it leads to image distortion, especially when dealing with the updates of the RGB color schemes. The SLSB working principle is that it conceals information in every pixel of only one color component in the RGB scheme. This means that SLSB will only hide information at each pixel on the same color component Red, Green, or Blue. The choice of the color to hide the information in is determined by an operation known as pair analysis that determines the color with the greater diversity ratio.

In order to achieve an enhanced image quality through applying filters algorithms, Gokul and his colleagues (Gokul *et al.* 2012) perceived SLSB as a more advanced technique that not only acts as a steganographic algorithm, but also as a form of cryptographic technique. Unlike the approach described by Warade and his colleagues (Warade *et al.* 2014), this method did not entirely use the RGB scheme but rather used

the image plane to decide where to embed the secret information. Gokul and his colleagues (Gokul *et al.* 2014) appended that the furthest pixel value is substituted with a unique number that represent what the authors call shares. Share represents the position in which the encryption will occur on the cover image. The extraction process involved the removal of the LSB pixels within the image cover to reveal the share position.

SLSB continues to gain popularity over traditional LSB due to its ability to improve clarity and reduce distortion. As Roque and his colleagues showed (Roque, 2013) various reasons that make the technique stand out among other spatial domain algorithms. SLSB, unlike other algorithms, is not subject to visual compromise primarily because it is difficult to perceive the hidden information with a human eye. Also, even if the attacker manages to reveal the LSB, it is difficult to discern the concealed information.

Warade and his colleagues (Warade *et al.* 2013) continued to add that, the strength of SLSB is further aggregated by the fact that there are no statistical operations that can be applied to the color pixel with the aim of identifying the hidden information. SLSB is relying on ratio analysis to determine the color scheme in which to hide the information. In some cases, it is even difficult to identify the difference between the original image and the one with the hidden information. SLSB is resistant to any comparison measures achieved through histogram analysis thereby making the two images indistinguishable.

Apart from encoding images, SLSB is also used to embed and conceal information in audio files. According to Priya and his colleagues (Priya *et al.* 2012), SLSB encoding is used to embed a special code in an audio file that conforms to digital format. In this

model, it is possible to encode an audio file to conceal a large amount of information than what can be concealed using a typical JPEG image. They added that SLSB coding for an audio file involves the substitution of the last two bits with exactly two bits of the message to be sent. However, they were quick to note that this method results in a situation whereby the audio file being used contained a lot noises. To reduce the noise content, they suggested that it is always wise to use a digital audio file with quality signal content that does not have much interference. Decoding the embedded message within the audio file involves accessing the index sequence of the process used to embed the message. However, this method distorts the audio file because the part that contains the embedded pixel bits tends to have lower pitches and tone.

SLSB is more tailored to address the security deficiencies evident in the LSB technique. Duvvuru and his colleagues (Duvvuru *et al.* 2014) proposed a 3-level reverse steganographic process aimed at boosting the security of the embedded process. The paper elaborated an experiment to show the effectiveness of the SLSB algorithm in implementing a multi-level steganographic security. In the experiment, SLSB was found to achieve the strongest security parameters when applied three times on a multi-level algorithm. However, the algorithm is often slow, and thus they suggested a combination of two SLSB and one LSB iteration of the algorithm phase to get optimal security results.

### Approach and Methodology

#### 3.1 Introduction

In this thesis, we present a new text hiding algorithm inside images. The algorithm is based on SLSB (selected least significant bit) substitution mechanism combined with three major enhancements: (fast encryption algorithm, randomization, and pixel selection to reduce hiding distortion).

#### 3.2 The Proposed Data Hiding Algorithm

The proposed algorithm is illustrated in Figure 5. This achieves the main desired features covered by steganography mechanism which are:

- Encryption complexity: this is important to reduce the hacking possibility, by uncompromising the data even if many extraction mechanisms are applied, since the data will be unreadable.
- Minimum distortion: to reduce the overall distortion, this results from the hiding process on the cover image. Keeping the difference between the original cover image and the modified one as minimum as possible. This is important to make the algorithm immune over stego-attacks, such as histogram analysis.
- Randomization: this approach is related to scatter the data in a manner that makes the extraction of data very hard.
- Good hiding capacity compared to distortion level: The algorithm is capable of hiding a maximum amount of data in trade of distortion.

The algorithm determines different starting point each time using a unique algorithm based on the blob detection algorithm which depends on the connected components labeling algorithm. The blob detection mechanism determines different objects inside the image using the binary version of the image, and continues to label each object using its edges. Our proposed algorithm selects the minimum or the maximum object size depending on user selection as start object. The center point for that object is then considered as the starting point for hiding the first bits.

The next pixel location to hide the data inside is calculated depending on the previous pixel location and two extra parameters predetermined by the sender. It is worth to mention, that the two parameters for the randomization function and the 3DES encryption are sent to the receiver over separated secure channel. The calculation of the next pixel location is illustrated using the formulas in equations 5 to 8.

Let  $w$  be the width of image,  $h$  is the height of the cover image,  $n$  is the number of objects extracted from blob algorithm of the cover image,  $c$  and  $d$  are constant integer numbers assigned by the user in a range of 1-  $\infty$ . Then we define the integer numbers  $a$  and  $b$  as:

$$a = [w/(n * c)] \quad (3)$$

$$b = [h/(n * d)] \quad (4)$$

Let  $(x_0, y_0)$  be the current pixel location then we define  $(x_1, y_1)$  to be the next pixel location as follows:

$$x_1 = (x_0 + a) \% w \quad (5)$$

$$y_1 = (y_0 + b) \% h \quad (6)$$

The modules operator is used to restrict the next selected pixels in the image boundaries. Moreover, the used equation guaranties that all image locations (pixels) will

be scanned without any repetition. This was proved experimentally using the implemented system. After that, we check the new location  $(x_1, y_1)$  against a specified threshold value of the selected color from the three colors component (R, G, and B). The thresholding process determines whether to use this pixel or not. This pixel is used to hide data if the color component value is equal or higher than the threshold. If the pixel does not pass the threshold, we generate a new pixel location. It is worth to mention that we use the threshold to reduce the distortion caused by hiding the data by applying the hiding mechanism only on the high intensity color components. Moreover, the selected color component used in hiding data is calculated by SLSB algorithm via neighbor matching mechanism by finding the color component with higher intensity variation between neighbors; i.e. the color with minimum distortion on the cover image.

After hiding the data, the proposed algorithm applies a smoothing function which alters the unused bits in the pixel to round the value of the whole pixel to be as close as the original one. Moreover, the smoothing function introduced by the algorithm not only apply the processing on the selected color component to make it as close as possible to the original color component, but continue to check the other color components in order to reduce the noise. An example of smoothing algorithm is shown in Table 4. The third column of Table 4 shows pixels after smoothing. Using this technique combined with high color intensity selection, we achieved the best noise reduction average in comparison with other published steganography algorithm as illustrated in the results analysis section.

Table 3: Smoothing algorithm example.

Original pixel		Pixel with hidden data		Pixel after smoothing	
G	B	G	B	G	B
1010 <b>0011</b>	11001010	1010 <b>0000</b>	11001010	1010 <b>0100</b>	11001010
1110 <b>1100</b>	11011011	1110 <b>1111</b>	11011011	1110 <b>1011</b>	11011011
1011 <b>1101</b>	1100 1011	1011 <b>1100</b>	1100 1011	1011 <b>1100</b>	<b>1111 1111</b>

The proposed algorithm is reversible, and we can extract the hidden data from the stego image. Figure 6 shows the data recovery block diagram for the proposed algorithm. The block diagram illustrates the steps for recovering the hidden data from the stego image depending on the received parameters from the sender which includes encryption key, randomization function constants (a and b), object type, and threshold value. Also, it shows that reconstructing data is done every four pixels as we use 2-bit SLSB data hiding which means for each character with eight bits four pixels are needed.

It is worth to mention here that, the presented approach does not have robustness for image processing methods like scaling, rotation, filtering, and object shifting. As these processes will change or even remove the embedded data inside the cover image. On the other hand, this cannot be considered as a weakness point since all the substitution mechanisms used on steganography cannot recover the full data after applying image processing techniques.

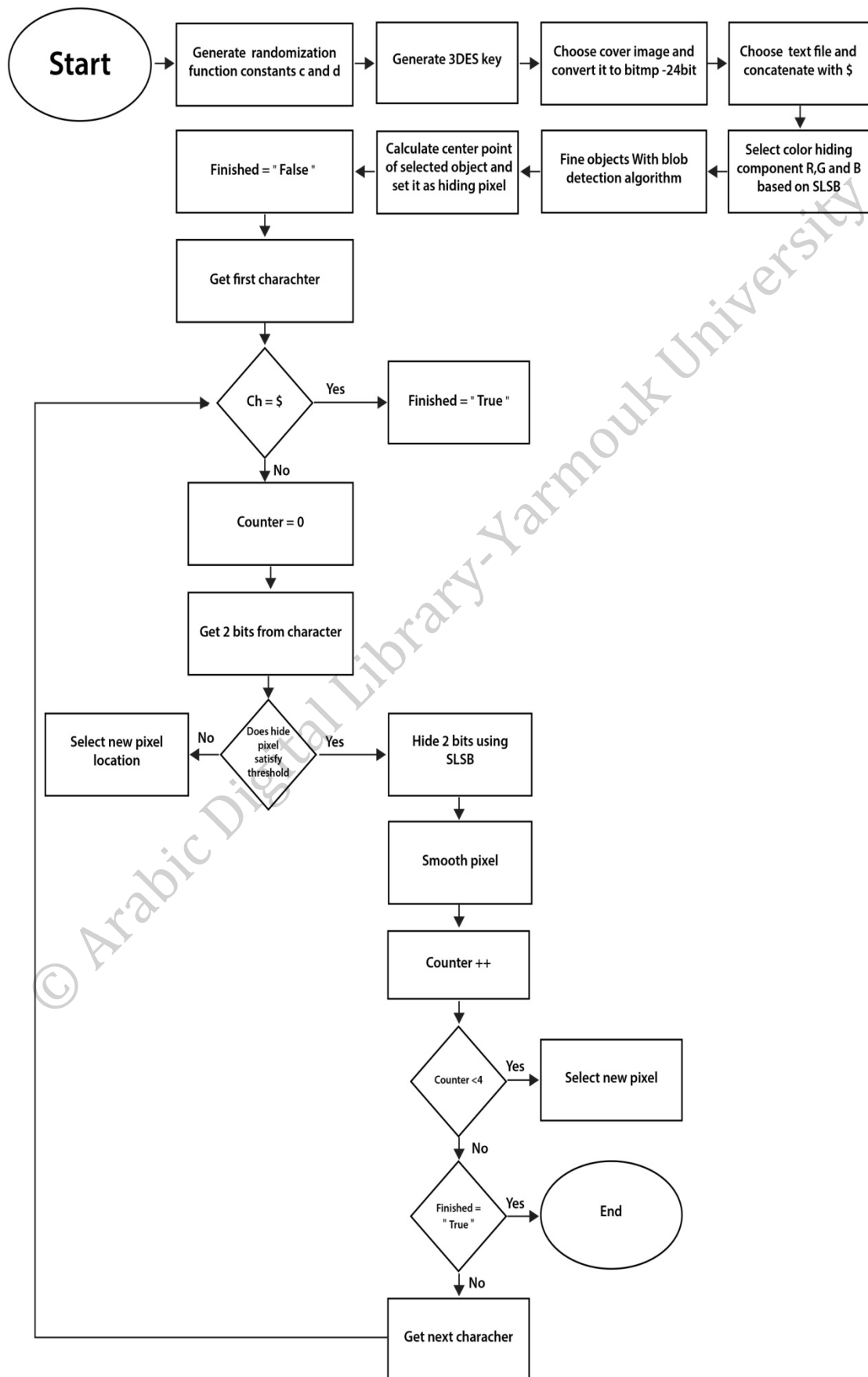


Figure 5: Block diagram for the presented algorithm sequence functions.



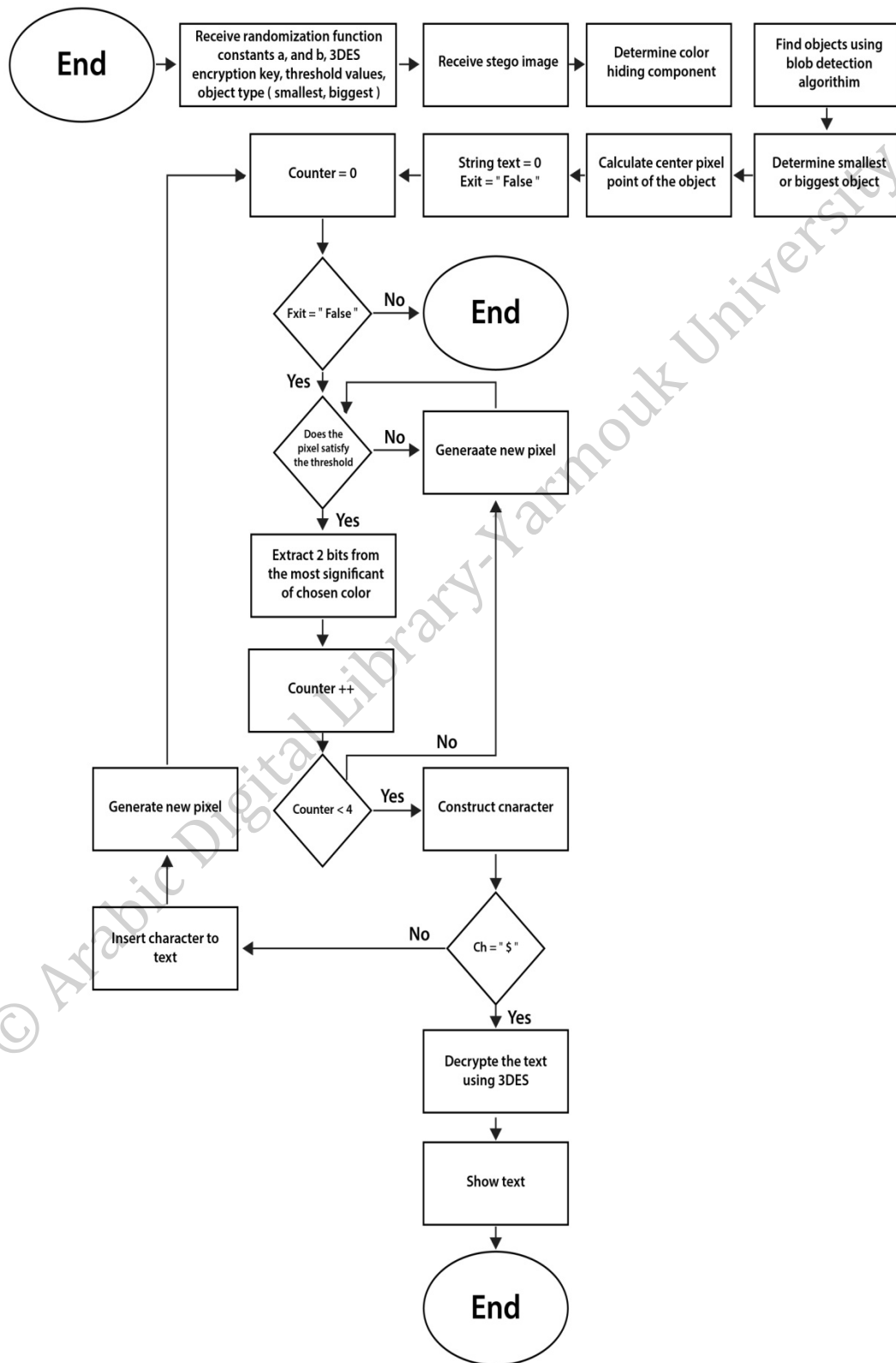


Figure 6: Block diagram for data recovering phase in presented algorithm.

### 3.3 Threshold Selection

The presented model introduces new mechanism to increase the complexity of data extraction, and reduce the noise on the stego image generated by the hidden data. This was done by applying a filter, which selects pixels with high intensity colors based on threshold selected by the user. Using this technique, we manage to scatter the data in non-sequential way, which increases the extraction complexity. Moreover, the different thresholds selection increases the parameters used in hiding the data, therefore increases the allover probabilities in case of stego attack.

Selecting the threshold is not an arbitrary. It must satisfy some rules to be considered as a valid threshold. These rules were introduced insure data recovery, smoothness, and the impact over the data hiding capacity. These rules can be summarized as follows:

- The least 3 bits of the threshold should be zero.

Since the presented model apply hiding the data only in the pixels which meet the threshold condition (greater or equal to), and because we adopted the 2-bit SLSB hiding mechanism. Then changing the data of the least two bits in the cover pixel will change the value of the color component. This color component could be missed by the recovery phase because it may be excluded by the condition if it changed to a lower value. Therefore, the least 2 bits had to be zeros to eliminate the number from being decremented by the hidden data.

Also, we need the third least bit of the threshold to be zero because it may be changed in the constructed stego image by the smoothing function which aims to reduce the noise effect via rounding the color component value to be as close as possible to the original one.

- The threshold number should be relative to the size and the color distribution of the cover image used for hiding the data and the number of characters being hidden. Even the higher threshold results in lower noise effect, it can't be chosen every time, since it has a reversible relation with the hiding capacity. So, it is needed to be chosen carefully according to the size of data needed to be hidden and the size of cover image being used. In General, the maximum image capacity for standard size of 1080\*720 can handle 194400 characters using SLSB 2 bits data hiding. On the other hand, it doesn't compromise image capacity; since it still can hide secret data. This will be explained in the experimental settings and dataset section. Therefore, the maximum achieved capacity will be less than SLSB 2 algorithm.

© Arabic Digital Library - Yamouk University

# CHAPTER FOUR

## Implementation and Discussion

### 4.1 Introduction

In this chapter, we evaluate the performance of our presented approach and provide the results and discussions of the performed tests and experiments. To compare the result of the proposed algorithm with published research, we selected a standard dataset to utilize it in the testing. After that, we compared our results with similar approached published recently.

### 4.2 System specifications for the developed tool

To develop the proposed algorithm, we used a personal computer that has the following specification:

- Intel® Core™ i5-6700T quad core Processor (3M Cache, up to 3.20 GHz).
- Built in Intel® HD Graphics card 530 with 1.7 GB of memory.
- 8 GB of Ram memory, type DDR3L with bus speed of 1333MHz.
- 1000 GB SATA3 bus speed Western Digital WD1002FAEX 64MB Cashed memory.

### 4.3 Software Tools

For the implementation, we utilized Microsoft C#.net programming language. Moreover, we utilized AFORG open source library, which contains ready functions

such as blob detection, localization, image filters, and many more features. Also, MATLAB software was used in order to compare our approach with the SLSB algorithm.

#### 4.4 Proposed Algorithm Implementation

To implement the proposed algorithm, a windows application was developed. The application allows the user to select image. The system then detects the objects, finds the smallest or largest one, and calculates the center point to start the hiding. The system also generates a random 3DES encryption key to be used for encrypting the data before hiding. Figure 7 shows the user interface of the developed application.

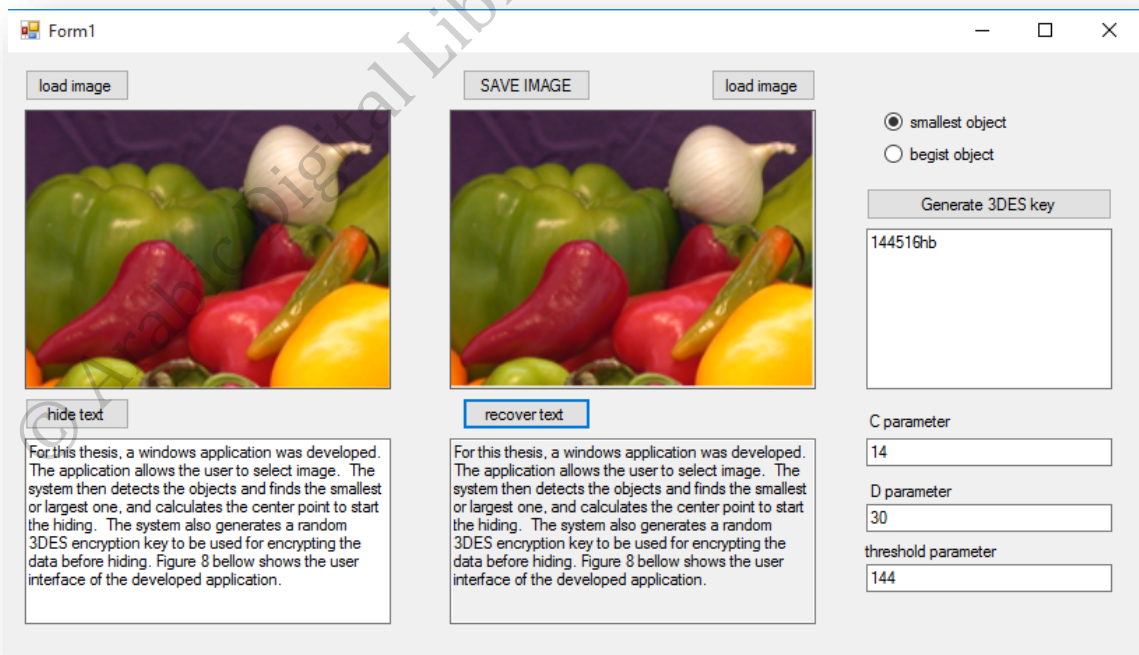


Figure 7: Developed system user interface.

As shown in Figure 7, the parameters  $c$  and  $d$  are integer values in average of  $1 - \infty$  as illustrated in equations (5) and (6) which is used to calculate the next position.

Threshold value is an integer between 144- 160 used to find the valid pixel for hiding data. Moreover, we implemented the data extraction algorithm in the same application to simplify testing and validation.

The application form in Figure 8 shows the hidden operations like calculating the smallest object labeled in red box which is extracted from the image. The figure also shows the randomly selected pixels for hiding the data in.

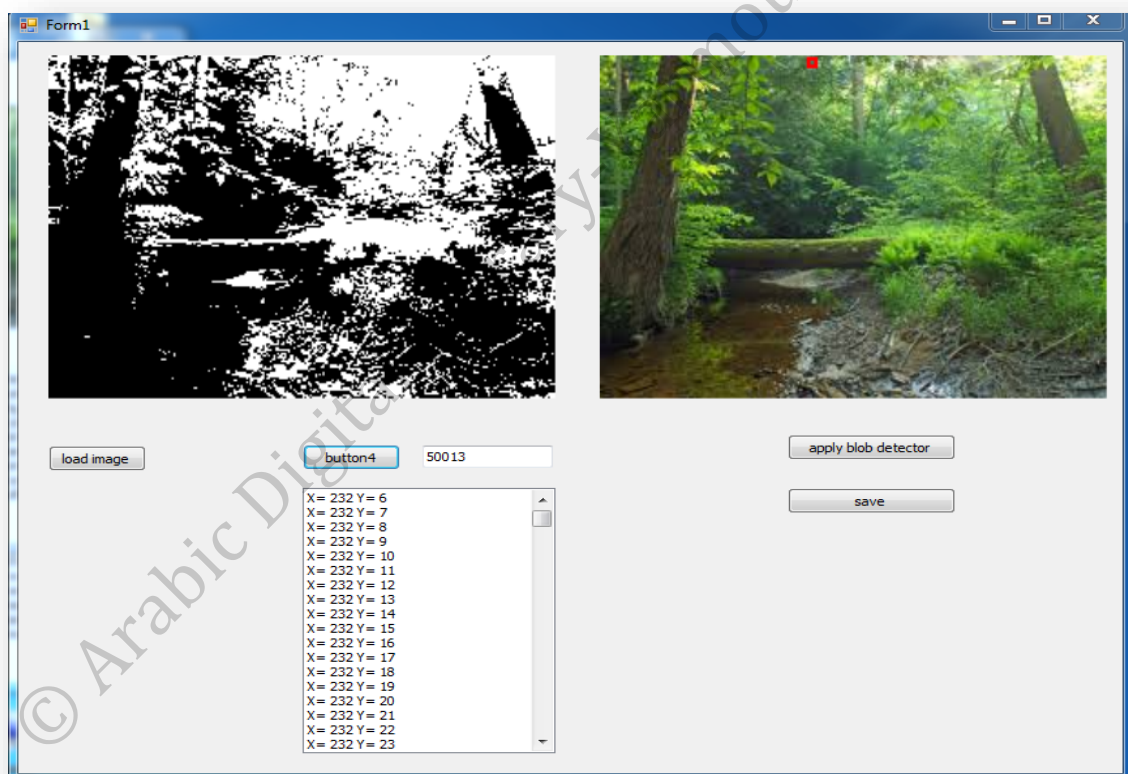


Figure 8: Blob detection and pixel randomization process.

For illustration, the implementation converts the cover image at first to binary image before extracting the objects. Later the smallest object is extracted and labeled in red, and the system calculates its center point as the first hiding pixel. The implementation confirmed that all pixels scanned in the image will be visited by the algorithm. Thus, the proposed algorithm will not reduce the capacity of data hiding.

#### 4.5 Experimental Settings and Dataset

In our experiments, we have used a dataset consisting of 20 colored images. The dataset was selected from standard defined images for image processing and image steganography papers like (Rahul & Kumar, 2012), (Ahmad *et al.* 2014), and others listed in the references. These images are used in most published image processing researches and registered software's like IP-Lap and MATLAB. Moreover, we used another images utilized by some published papers for comparison purposes. The selected dataset covers the parameters we are targeting in our research illustrated bellow:

- Size: The dataset covers different sizes from very small scales 150 \*178 to HD scale of 1024 \* 720. This variation was needed to test the capacity of the presented algorithm.
- Color intensity: Colors distribution in images is a vital factor in the presented algorithm since we depend on SLSB to select the best color component with least effect over the whole image. Moreover, the selected color histogram has to be concaved to the right to enable the maximum data hiding capacity, as we use higher thresholds. Therefore, the dataset must contain different colors intensity to test all the distribution cases.
- Resolution: Image resolution also plays a main factor in the proposed algorithm since the lower image resolution reflects higher noise in the stego image. Therefore, lower image resolution has to be included in the test to evaluate the performance of the algorithm.

It is worth to mention that the algorithm converts each input image into 24-bit pixel resolution. This conversion does not affect the resolution since it only adds the missing bits with zero values to the left of each color component.

Table 4: Dataset image distribution based on size.

Image size / pixel	Number of Images
150*178	2
276*183	2
481*359	3
640*337	4
512*512	4
670*430	1
1024*720	4




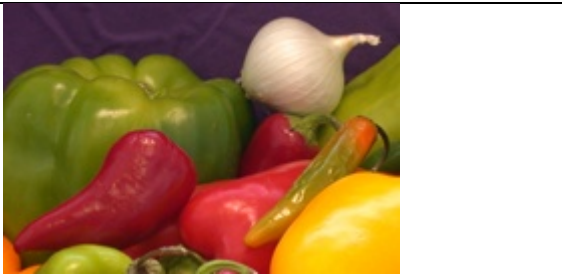
Table 4 below shows the dataset image distribution over the different sizes (defined by pixels number in each image).


Table 5 below shows some of the dataset images used in our evaluation, with the size and type of each one. The table shows the different image sizes and color intensity variation.

Table 5: Sample of images used for testing the proposed algorithm.

No	Name	Image	Size	Type
1	People		431*359	.bmp



2	Tulips		1024*768	.bmp
3	Koala		1024*768	.bmp
4	People2		276*183	.bmp
5	Onion		198*135	.bmp

No	Name	Image	Size	Type
6	Lenna		512*512	.bmp

For thresholding, we have adopted several approaches to decrease the noise on the stego image, and increase the extraction complexity. This is because different thresholds mean different pixels selection for data hiding. We studied different threshold ranges on the PSNR values of the stego images. This was done empirically by implementing different threshold values over the same images, and calculates the overall PSNR average. Then we manage to find the best ranges without losing much hiding capacity from the cover image. Table 6 shows the different threshold sets used for testing with their average PSNR.

Table 6: Threshold sets used to determine the best ranges for algorithm threshold.

	Images					Average PSNR (dB)
	1	2	3	4	5	
Threshold values	PSNR values (dB)					
26	33.864	43.926	55.934	27.712	28.961	38.0794
40	33.961	44.864	56.624	27.523	29.541	38.5026
56	34.9421	44.4962	57.7062	27.115	29.2778	38.7074

72	37.0421	46.5962	59.8062	29.215	31.3778	40.8074
96	38.8421	48.3962	61.6062	31.015	33.1778	42.6074
120	40.9421	50.4962	63.7062	33.115	35.2778	44.7074
144	42.8421	52.3962	65.6062	35.015	37.1778	46.6074
160	44.1421	53.6962	66.9062	36.315	38.4778	47.9074
200	44.9421	54.4962	67.7062	37.115	39.2778	48.3074
224	45.2421	54.7962	68.0062	37.415	39.5778	48.5074

Figure 9 shows the threshold values with respect to average PSNR. It illustrates that the values of threshold after 200 has small effect over the PSNR.

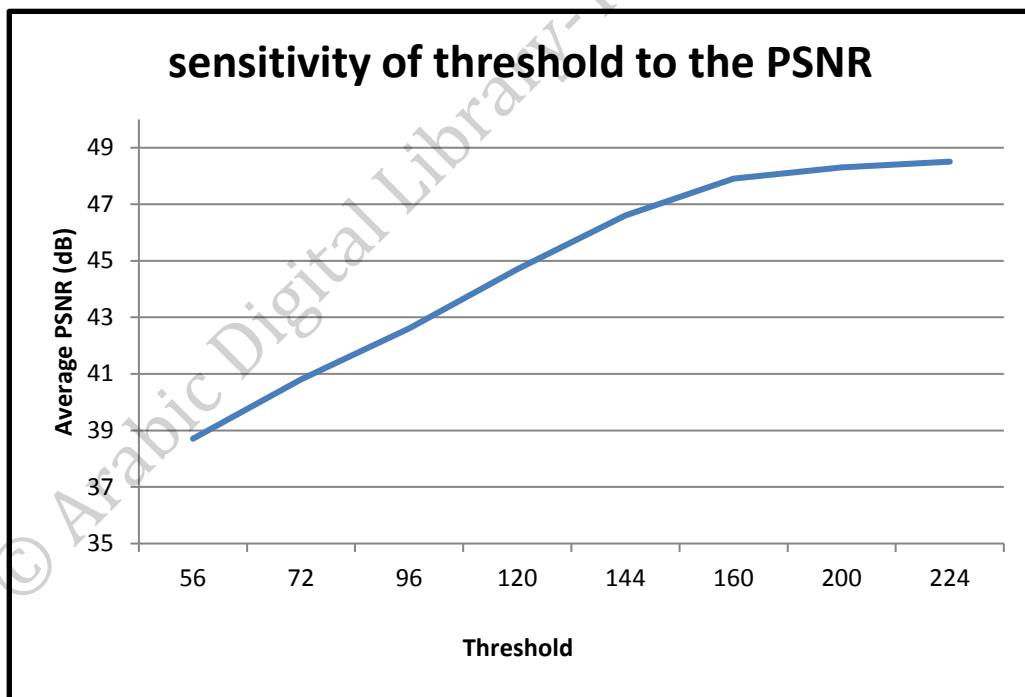


Figure 9: Average PSNR with respect to threshold value.

Also, we studied the hiding capacity with respect to the threshold values. The results are clarified using Table 7 which shows the average capacity hiding with respect to threshold value. Figure 10 views the average of hiding capacity with respect to the

threshold. Combining both results of PSNR and average capacity to select the best threshold value, we concluded that the range of 144 to 160 is the best range.

Table 7: Average capacity for the test images over different thresholds.

Threshold Values	Hiding Capacity (Character)
26	5346.25
40	5146.71
56	5046.13
72	4987.81
96	4973.53
120	4753.99
144	4345.16
160	4098.28
200	4003.41
224	3964.58

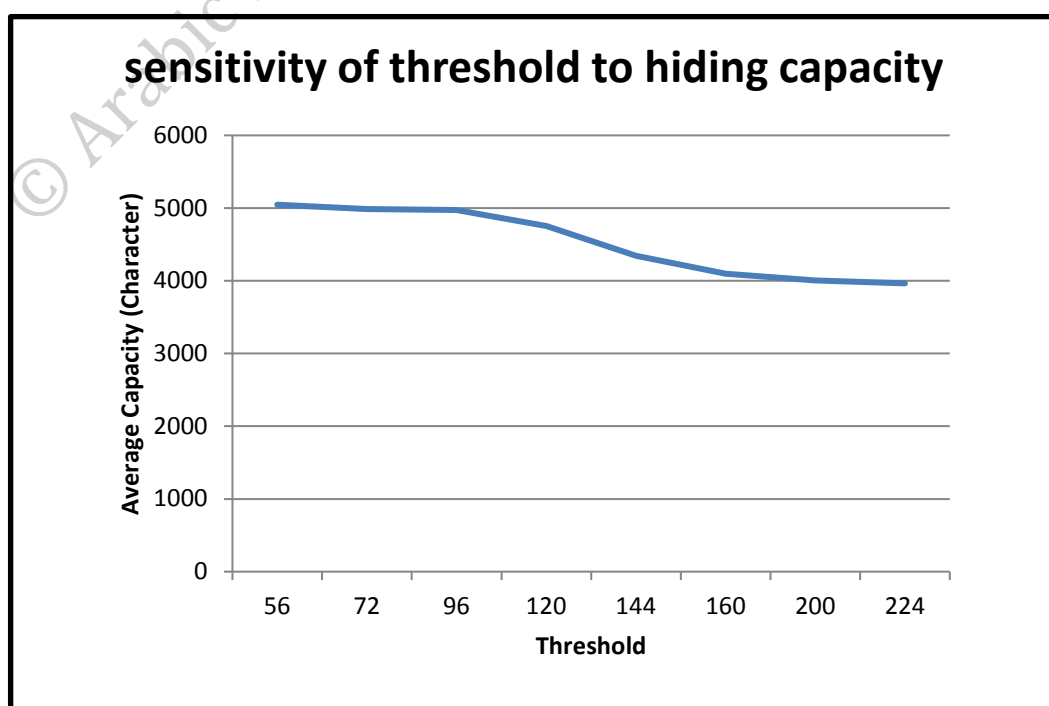


Figure 10: Threshold sets sensitivity curve according to their PSNR results.

#### 4.5 Experimental Results

The proposed algorithm is implemented and the MSE, SNR, and PSNR are measured. Also, the same experiment is conducted using the same images for SLSB algorithm. Table 8 shows the obtained results and proves the proposed algorithm efficiency is better than the original SLSB.

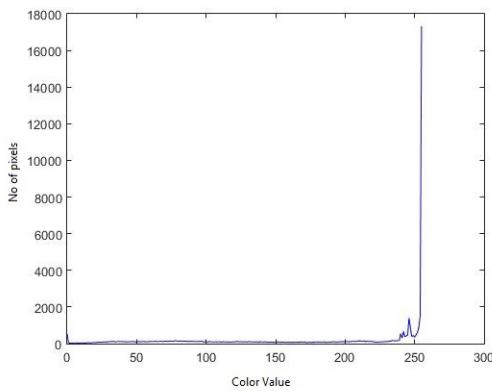
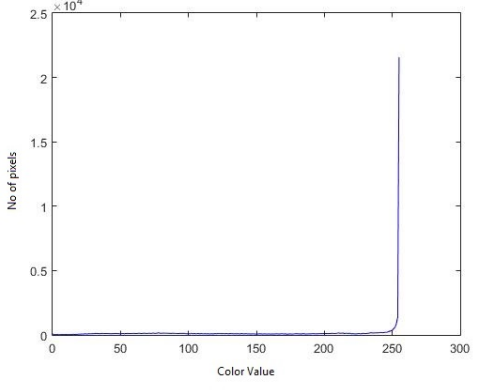
Table 8, shows four stego images outputs from SLSB and the presented algorithm. It views the MSE, SNR and PSNR. Moreover, histogram analysis was carried out. It illustrates the distribution of data over the color intensity and clarifies the enhancement of the presented model over the standard SLSB.

The histogram shows the massive reduction of changes over the original pixels compared to the original one for the presented model. It views the weakness of the SLSB standard algorithm.

Table 8: Comparison between presented algorithm and SLSB.

SLSB				presented algorithm			
Image name	MSE	SNR	PSNR	Image name	MSE	SNR	PSNR
People 1	104.4397	25.3662	27.9421	People 1	66.4397	43.9662	44.1421

 <p>Figure 1: Histogram for SLSB algorithm. The x-axis is 'Color Value' (0 to 300) and the y-axis is 'No of pixels' (0 to 18000). A sharp peak is visible at color value 250, reaching a height of approximately 17000 pixels.</p>	 <p>Figure 2: Histogram for presented algorithm. The x-axis is 'Color Value' (0 to 300) and the y-axis is 'No of pixels' (0 to 2.5 x 10^4). A sharp peak is visible at color value 250, reaching a height of approximately 2.2 x 10^4 pixels.</p>
---	---

SLSB				presented algorithm			
Image name	MSE	SNR	PSNR	Image name	MSE	SNR	PSNR
Tulips	20.5807	30.6276	34.9962	Tulips	14.3807	48.6376	53.6962
Lenna	77.1494	24.1199	29.2575	Lenna	59.4494	44.4199	46.3975
Onion	245.6408	17.2209	24.2278	Onion	194.4408	33.3709	38.4778

## 4.6 System evaluation

For the system evaluation, a sequence of tests has been executed on the defined images to extract the tests metrics explained in the methodology section (MSE, SNR, and PSNR). Six images were adopted and the results for the MSE, SNR and PSNR were calculated as shown in Figure 11, the average MSE was 40.8 which considered a very low error for the image full capacity hiding.

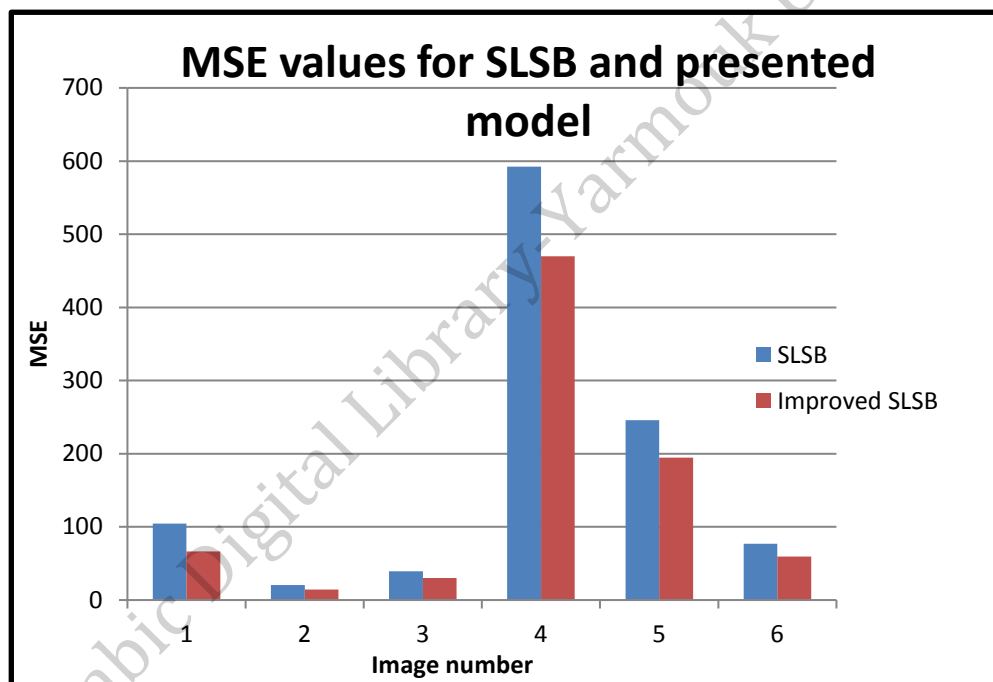


Figure 11: Calculated results of MSE for the proposed algorithm and the SLSB one.

Figure 12 shows the SNR for 6 images. The average SNR was 18.3633dB. This is considered a very good result compared to other proposed algorithms in terms of full image capacity usage. Figure 12 shows the standard SLSB result for the same hiding capacity, which indicates a lower SNR average value.

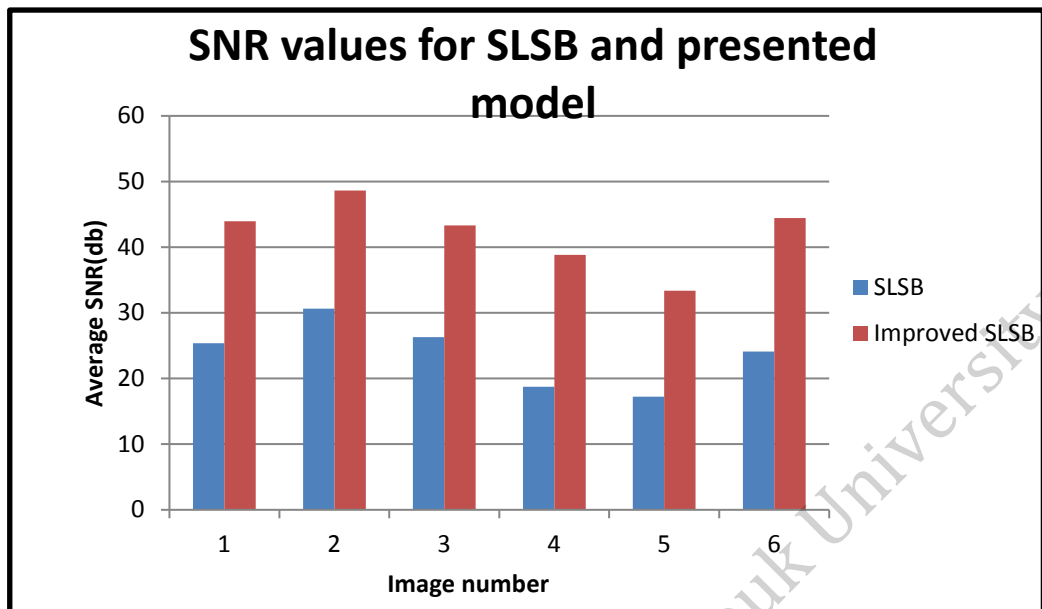


Figure 12: SNR comparison.

Third experiment to measure PSNR was conducted. Figure 13 shows the PSNR of the six images and illustrate the average PSNR of 15.901dB. This is a very good value in terms of different image size and resolutions. Figure 13 also shows the standard SLSB average PSNR over the same images and data, with a lower PSNR average compared to the presented model.

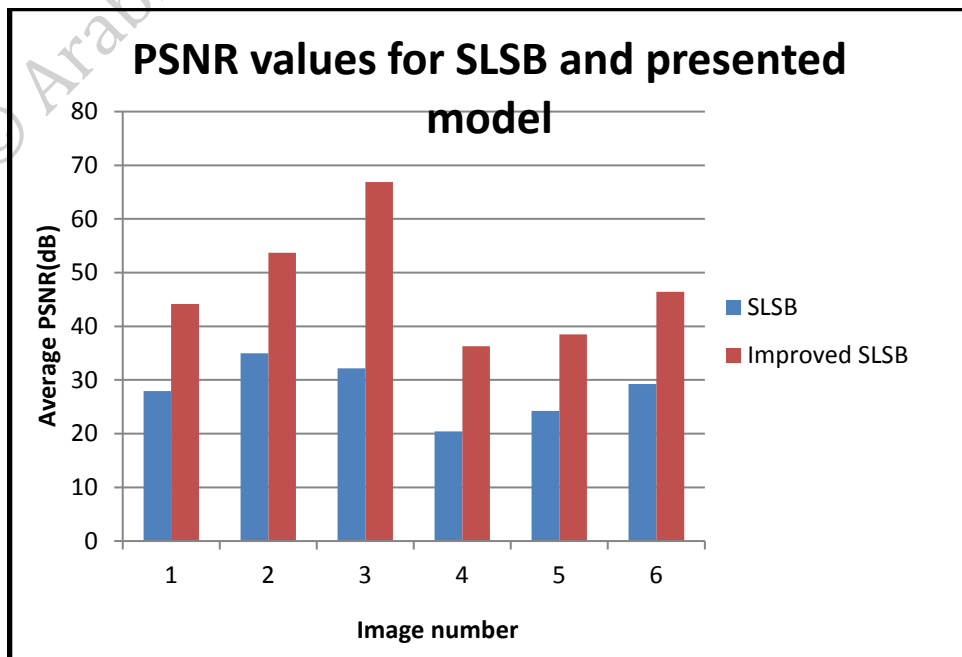


Figure 13: PSNR comparison.



For testing time efficiency of the system, we had calculated the time needed for hiding 4,000 characters using the proposed algorithm. Table 9 shows the execution time for the hiding and recovering phase over the test dataset. The Table illustrates that the time needed for both phases are very limited with average of 370.75 milliseconds.

Table 9: Proposed model hiding and recovery phases execution time.

Image number	Image name	Hiding time/ milliseconds	Recovery time/ milliseconds
1	People	180	108
2	Tulips	1530	31
3	Koala	1542	32
4	People 2	343	29
5	Onion	69	30
6	lenna	519	36
Average time/ milliseconds		697.17	44.33
Average total time/ milliseconds		370.75	

Moreover, we applied Kolmogorov-Smirnov test (KS-test) to clarify the differences between the proposed algorithm and the SLSB algorithms. The KS-test is a statistics test used to determine if two datasets differ from each other clearly. It is a Non-parametric and distribution free, making no assumption about the distribution of data. KS-test test was developed to decrease dataset to a few simpler statistics numbers (mean, median, high, low, standard deviation).

The Kolmogorov-Smirnov (K-S) test is based on the empirical distribution function (ECDF). Given  $N$  ordered data points  $Y_1, Y_2, \dots, Y_N$ , the ECDF is defined as:

$$EN=n(i)/N \quad (6)$$

Where  $n(i)$  is the number of points less than  $Y_i$  and the  $\{Y_i\}$  are ordered from smallest to largest value. This is a step function that increases by  $1/N$  at the value of each ordered data point (Kolmogorov-Smirnov Goodness-of-Fit Test).

We accumulate the summation of values for each SLSB and the presented model. Figure 14, 15, and 16 present the K-S test for MSE, SNR, and PSNR respectively for both SLSB and the presented model. It shows the advance of the presented model on the SLSB.

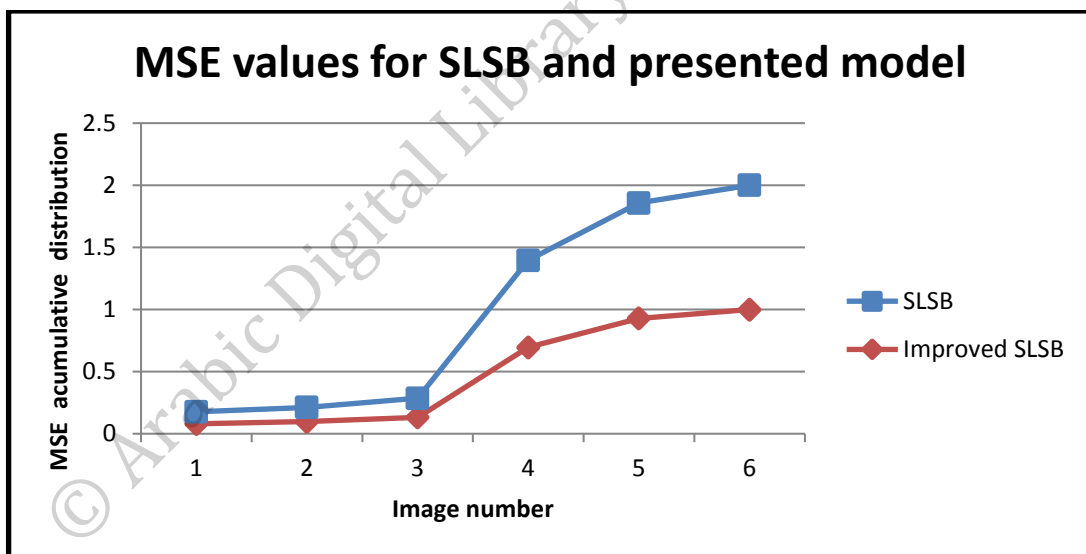


Figure 14: MSE K\_S test Comparison

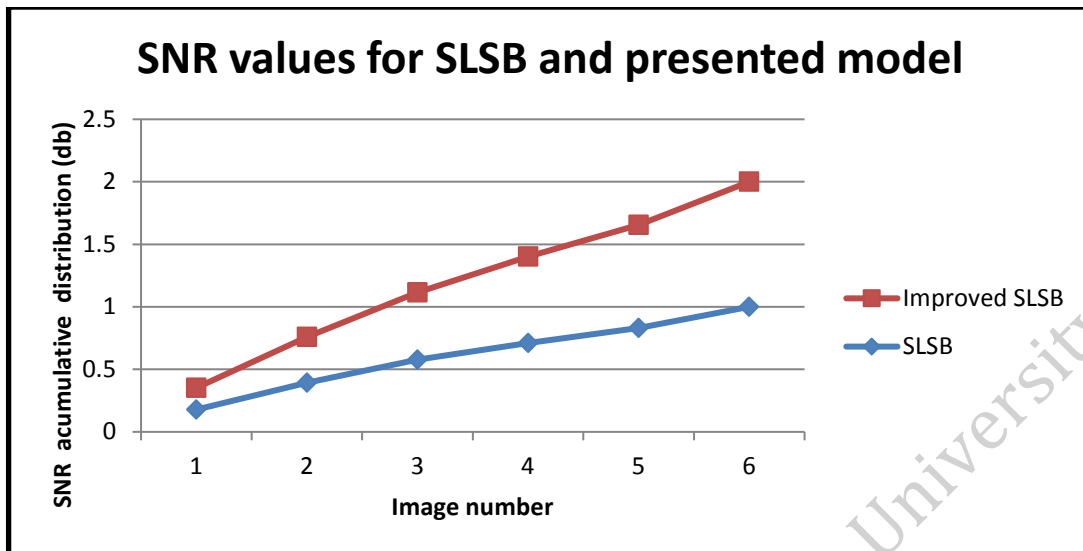


Figure 15: SNR K\_S test Comparison.

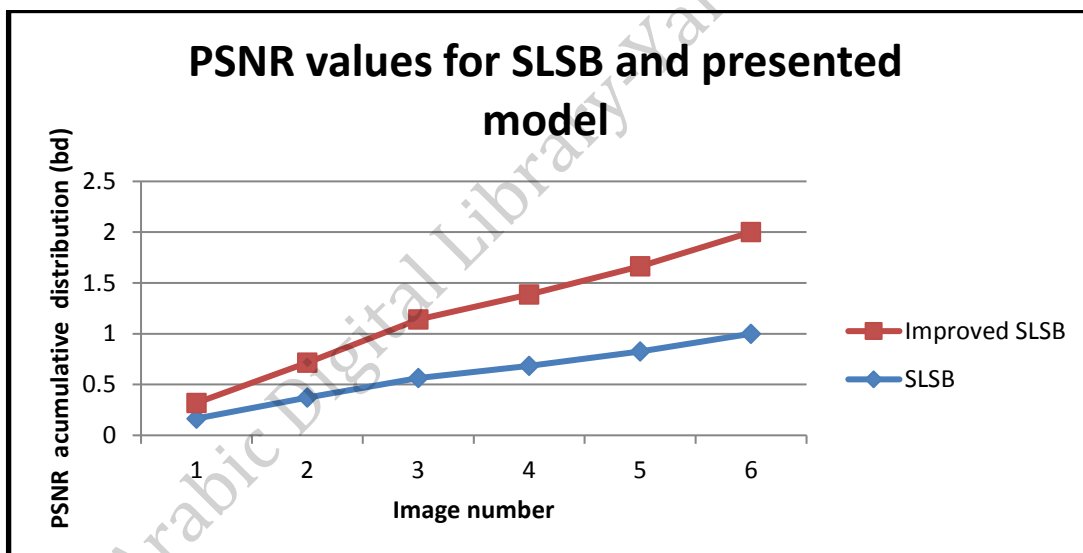


Figure 16: PSNR K\_S test Comparison.

### Conclusions and future works

In this chapter, we introduced conclusions, achievements and the recommended future works.

#### 5.1 Conclusion

In this thesis, new steganography mechanism targeted to hide text inside images based on SLSB substitution technique was presented. The presented approach reduces the noise caused by the hidden data through suggesting new pixel selection and smoothing technique. Also, we imposed a randomized distribution of data to harden the data extraction. The algorithm also uses symmetric encryption to raise the security with a shared key sent on standalone channel. The results show the efficiency of the system compared to the original SLSB algorithm with higher PSNR and SNR values than the original SLSB algorithm with average of 15.901dB, 18.3633dB consequently. Moreover, the proposed algorithm achieved less mean square error with average of 40.8 compared to the SLSB results. The algorithm also increased the extraction complexity since it added new parameters that need numerous probabilities to be hacked via stego analysis mechanisms.

#### 5.2 Future work

As a future work, we will generate a web services version of the system to enable the use of this methodology over the web services, also an extra enhancement will be added

to the algorithm by applying a multi-level hiding over the same presented approach. New start point selection mechanism will be offered to increase the parameters probabilities. This will extremely harden the extraction complexity.

We suggest developing more mature version of the system that exports the used keys in the hiding phase as encrypted XML file to be sent over secure tunnels in separate and standardized way.

© Arabic Digital Library-Yarmouk University

## References

- Rahul , J., & Kumar, N. (2012). Efficient data hiding scheme using lossless data compression and image steganography. *International Journal of Engineering Science and Technology* , 3908-3915.
- Roque, J. J., & Minguet, J. (2013). *SLSB: Improving the Steganographic Algorithm LSB*. Retrieved from Faculty of Engineering- De La Republica- Uruguay: [http://www.fing.edu.uy/inco/eventos/cibsi09/docs/Papers/CIBSI-Dia3-Sesion9\(1\)](http://www.fing.edu.uy/inco/eventos/cibsi09/docs/Papers/CIBSI-Dia3-Sesion9(1))
- Ahmad, J., Muhammad, K., Rehman, N., Jan, Z., & Qureshi, R. (2014). A Secure Cyclic Steganographic Technique for Color Images using Randomization. *Technical Journal* , 354-357.
- Babita, A., & Kaur, M. (2009). High Capacity Filter Based Steganography. *International Journal of Recent Trends in Engineering* , 672-674.
- Bellare, M., & Rogaway, P. (1995). *Optimal Asymmetric Encryption: How to Encrypt with RSA*. University of California, USA.
- Cameron, R., & Wyler, N. R. (2007). *Juniper Networks Secure Access SSL VPN Configuration*. Syngress.
- Champakamala, B., K., P., & Radhika, D. (2012). Least Significant Bit algorithm for image steganography. *International Journal of Advanced Computer Technology* , 34-38.
- Chang, C. C., Wang, Z. H., & Yin, Z. X. (2009). An Ingenious Data Hiding Scheme for Color Retinal Image. *Proceedings of the Second Symposium International Computer Science and Computational Technology*, (pp. 1-6). China.
- Cisco SSL Appliances*. (2017). Retrieved from Cisco: <https://www.cisco.com/c/en/us/products/security/ssl-appliances/index.html#datasheets-literature>

- Davis, T. (2000). *www.geometer.org/mathcircles*. Retrieved from [www.geometer.org](http://www.geometer.org/mathcircles):  
<http://www.geometer.org/mathcircles>
- Duvvuru, R., Rao, J., Singh, S., & Suman, R. (2014). Performance Analysis of Multi-class Steganographic Methods Based on Multi-Level Re-steganography. *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India*, 535-542.
- Filler, T., Judas, J., & Fridrich, J. (2011). Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes. *IEEE Transactions on Information Forensics and Security*, 920-935.
- Fridrich, J., Goljan, M., & Soukal, D. (2006). Wet paper codes with improved embedding efficiency. *IEEE Transactions on Information Forensics and Security*, 102-110.
- Gaikwad, D. P., & Wagh, S. (2010). Colour Image Restoration For An Effective Steganography. *I-manager's Journal on Software Engineering*, 65-71.
- Gokul, M., Umeshbabu, R., Shriram, V. K., & Deepak, K. (2012). Hybrid Steganography using Visual Cryptography and. *International Journal of Computer Applications*, 5-8.
- Handschuh, H., Lucks, S., Preneel, B., & Rogaway, P. (2007). *tream ciphers*, Ruhr-Universität Bochum. Retrieved from *tream ciphers*, Ruhr-Universität Bochum:  
<http://www.drops.dagstuhl.de/opus/volltexte/2009/1959>.
- Ibrahim, A., Zabian, A., Esteteya, F. N., & Al padawy, A. K. (2009). Algorithm for Text Hiding in Digital Image for Information. *IJCSNS International Journal of Computer Science and Network Security*, 262-268.
- Inzunza-González, E., & Cruz Hernández, C. (2009). Software to encrypt messages using public-key cryptography. *World Academy of Science, Engineering and Technology*, (pp. 623-627).

- Kahate, A. (2013). *Cryptography and Network Security*. Tata McGraw-Hill Education Pvt. Ltd.
- Kaur, G., & Kaur, K. (2013). Digital Watermarking and Other Data Hiding. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* , 181-183.
- Kaza, C. (2006). *Least Significant Bit Steganography Detection with machine Learning Techniques*.
- Ker, A. (2005). Steganalysis of LSB Matching in Grayscale Images. *IEEE Signal Processing Letters* , 441-444.
- Kolmogorov-Smirnov Goodness-of-Fit Test*. (n.d.). Retrieved from Engineering Statics Handbook.
- Kolmogorov-Smirnov Test*. (n.d.). Retrieved august 1, 2017, from <http://www.physics.csbsju.edu>.
- Latika, & Gulati, Y. (2015). A Comparative Study and Literature Review of. *IJSTE - International Journal of Science Technology & Engineering* , 238-241.
- Medeni, O. M., & Souidi, M. (2010). A Generalization of the PVD Steganographic Method. *International Journal of Computer Science and Information Security* , 156-159.
- Mstafa, J. R., & Khaled , M. E. (2015). An Efficient Video Steganography Algorithm Based on BCH Codes . *ASEE Northeast Section Conference* .
- Mstafa1, R., & Bach, C. (2013). Information Hiding in Images Using. *ASEE Northeast Section Conference* .
- Ni, Z., Shi, Y., Ansari, N., & Su, W. (2008). Robust Lossless Image Data Hiding Designed for Semi-Fragile Image Authentication. *IEEE Transactions on Circuits and Systems for Video Technology* , 497-509.
- Poweski , B., & Raphael, D. (2009). *Security on Rails*.



- Pradeep, K. P., & Devendra, K. (2012). Security through SSL. *International Journal of Advanced Research in* , 178-184.
- Priya, S., Mahesh , K., & Kuppusamy, K. (2012). Efficient Steganography Method to Implement Selected Least Significant Bits in Spatial Domain. *International Journal of Engineering Research* , 2248-9622.
- Roque,, J. J., & Minguet, J. M. (2010). *SLSB: Improving the Steganographic Algorithm LSB*. Spain: Universidad Nacional de Educación a Distancia.
- Sarkar, A., Madhow, U., & Manjunath, B. (2010). Matrix Embedding With pseudorandom Coefficient Selection and Error Correction for Robust and Secure Steganography. *IEEE Transactions on Information Forensics and Security* , 225-239.
- Shahana. (2013). A Secure DCT Image Steganography based on Public-Key. *International Journal of Computer Trends and Technology (IJCTT)* , 2039-2043.
- Simmons, G. (1979). *Symmetric and Asymmetric Encryption*. New Mexico: Sandm Laboratories, Albuquerque.
- Siper, A., Farley, R., & Lombardo, C. (2005). *The Rise of Steganography*. New York, USA: Pace University.
- Su, P., & Kuo, C. (2003). Steganography in JPEG2000 compressed images. *IEEE Transactions on Consumer Electronics* , 824-832.
- Sumathi, C., Santanam, T., & Umamaheswari, G. (2013). A Study of Various Steganographic Techniques Used for Information Hiding. *International Journal of Computer Science & Engineering Survey* , 9-25.
- Thiyagarajan, P., Natarajan, V., Aghila, G., & Venkatesan, P. (2013). Pattern based 3D image Steganography. *3D Research* , 83-89.

- Wahaballa, A., Wahballa, O., Ramadan, M., & Qin, Z. (2014). Multiple-Layered Securities Using Steganography and Cryptography. *International Journal of Computers and Applications* , 93-100.
- Warade, S., Tijare, P., & Sawalkar, S. (2014). Data Security Using Cryptography and SLSB Algorithm. *International Journal of Research in Advent Technology* , 354-357.
- Wikipedia. (2017). *Steganography*. Retrieved from Wikipedia, the free encyclopedia: <https://en.wikipedia.org/wiki/Steganography>
- Xinpeng , Z., & Shuozhong , W. (2005, January 1). Steganography using multiple-base notational system and human vision sensitivity. *IEEE* , pp. 67-70.
- Yang, C., Weng, C., Wang, S., & Su, H. (2008). Adaptive Data Hiding in Edge Areas of Images with Spatial LSB Domain Systems. *IEEE Transactions on Information Forensics and Security* , 488-497.

© Arabic Digital Library - Yamouk University

## الملخص

صفاء عكور، آلية محسنة لإخفاء البيانات اعتماداً على عشوائية التبديل لطريقة SLSB، ماجستير هندسة الحاسوب، أتمتة صناعية، قسم هندسة الحاسوب في جامعة اليرموك، 2018، الدكتور المشرف: (د. محمد الجراح)

أصبح تنقل المعلومات من أهم متطلبات الحياة، وذلك بسبب تطور الانترنت والطلب المتزايد على الشبكة العالمية. حيث أصبحت الحوسبة السحابية نطاقاً مهماً لخدمات الانترنت الاجتماعية والمالية والعسكرية وإدارة الأعمال وغيرها. ونظراً لأهمية حماية المعلومات وخصوصاً مع استخدام بروتوكولات الإرسال، العديد من الدراسات بحثت في مجال المعلومات مثل السماح بالإطلاع عليها، تشفيرها، إخفاؤها، أو حتى تصحيحها.

في هذه الأطروحة نركز على إخفاء البيانات (Steganography) لأغراض حماية المعلومات المتراسلة. ومع تزايد البحث عن إخفاء المعلومات يزداد مستوى حمايتها. تقدم هذه الأطروحة طريقة مُحسنة لطريقة (SLSB) لإخفاء البيانات داخل الصور. وتستخدم هذه الطريقة أسلوب الكشف عن النقطة لتكون نقطة البداية لإخفاء المعلومات. ثم تقوم بعمل مسح شبه عشوائي لإخفاء المعلومات. نستخدم في هذه الطريقة إقتراناً ليحدد عشوائياً ثوابت من صورة الغلاف، و يُختار لون من الألوان الثلاثة (الأحمر، أو الأصفر، أو الأخضر) في بداية العملية اعتماداً على إحصائيات الألوان للصورة التي تحمل المعلومات. تستخدم الطريقة أيضاً آلية (3DES) لتشفير المعلومات قبل إخفائها في صورة الغلاف. في هذه الطريقة أيضاً البيكسل الذي لا يلائم حد معين ومحدد لا يُستخدم للإخفاء ويتم البحث عن بيكسل آخر يُحقق المطلوب.

نتائج التجربة للأطروحة أظهرت تحسناً على الأداء والفعالية مقارنةً مع طريقة SLSP. حيث حققت هذه الطريقة نسب أعلى في معدلات PSNR لتصل إلى (dB 15.901) ونسب أعلى في SNR (dB 18.3633) مقارنةً مع طريقة SLSB الأصلية. وكما حققت أيضاً هذه الطريقة المُحسنة قيم أقل في معدلات الخطأ MSE لتصل إلى (40.8) مقارنةً مع طريقة SLSB. وليس ذلك فحسب، بل حققت أيضاً نتائج أكبر في مدى صعوبة إستخراج المعلومات وذلك لإستخدام متغيرات مختلفة. كل هذا يجعل طريقتنا أكثر أماناً من إستخدام SLSB.

كلمات مفتاحية: تنقل المعلومات، Steganography، إخفاء البيانات، SLSB، تشفير، كشف نقطة.