International Journal of Advanced Computer Science and Information Technology (IJACSIT)
Vol. 2, No. 4, 2013, Page: 111-131, ISSN: 2296-1739

www.elvedit.com

Image Encryption Using Lagrange-Least Squares Interpolation

Authors

Mohammed A. Shreef
Informatics Institute for Postgraduate Studies/ Iraqi Commission for Computers and Informatics
skymood1987@gmail.com
Baghdad, 10001, Iraq

Haider K. Hoomod
Al-Mustansirya University-College of Education/Computer Science Department
drhjnew@gmail.com
Baghdad, 10001, Iraq

Abstract

Today, information security is becoming one of the most important issues in social network era. The fast development of network technology leads to facilitate many aspects of life, but it also gives attackers or unauthorized users an opportunity to violate the privacy of people. Encryption is a common technique that exists to protect information security, thereby deters attackers. Actually, digital images are widely used in storage and communication applications. Therefore, the protection of image data from unauthorized access has attracted much attention recently. This paper adopts a new image cryptosystem, XLLS, which consists of two main parts: encryption/decryption algorithm and ciphered key. The encryption algorithm is composed of two main stages: the diffusion stage and the substitution stage. In the diffusion stage, the pixels values are modified so that a slight change in one pixel is spread out to all pixels in the image. This stage completely depends in its construction on 'XOR' operation. For the substitution stage, it mainly composes of two encryption processes: Lagrange Process (LP) and Least Squares Process (LSP). This stage aims at changing the value of each pixel in the diffused image by using the principles of Lagrange interpolation and least squares method. For the decryption algorithm, it is simply the reverse of the encryption algorithm. On the other hand, the proposed cryptosystem introduces two different approaches of initial key. The users have option to choose any one of them to encrypt the plain-image. In the first approach, the proposed cryptosystem uses a key whose length is of 192 bits (24 bytes) in hexadecimal system as its input, and then expands it by using AES-192 key expansion algorithm. Conversely, in the second approach, the proposed cryptosystem uses an image as a key to cipher the plain-image, and then processes and expands the key-image by using the CBI key expansion algorithm.

Key Words

Image Encryption, Lagrange, Least Squares.

I. INTRODUCTION

In recent days of Internet, the encryption of data plays a major role in securing the data in online transmission and focuses mainly on its security across the internet. Different encryption techniques are used to protect the confidential data from unauthorized use.

Encryption is a very common technique for promoting the image security. The main idea in the image encryption is to transmit the image securely over the network so that no unauthorized user can decrypt the image. The image data have special properties such as bulk capacity, high redundancy and high correlation among the pixels that impose special requirements on any encryption technique [1].

Section two of this paper explains the classification of encryption algorithms, while mathematical preliminaries for the proposed cryptosystem are explained in section three. Section four and section five present the proposed cryptosystem and the proposed encryption techniques, respectively. Section six shows the implementation and experimental results. Finally section seven shows the conclusions.

II. CLASSIFICATION OF ENCRYPTION ALGORITHMS

Encryption algorithms can be classified in different ways; according to structures of the algorithms, according to keys, or according to the percentage of the data encrypted [2].

A. Classification According to Encryption Structure

Encryption algorithms can be classified according to encryption structure into block ciphers and stream ciphers [2].

A block cipher is a type of symmetric-key encryption algorithms that transforms a fixed-length block of plain-text data into a block of cipher-text data of the same length. The fixed length is called the block size. For many block ciphers, the block size is 64 or 128 bits [2].

In contrast to block ciphers, which operate on large plain-text blocks, **stream ciphers** operate on smaller units of data at a time. Typically, a random bit stream is required to serve as a key-stream. It is then XORed with the plain-text stream to accomplish the encryption process [3].

B. Classification According to Keys

Encryption algorithms can be classified according to keys into two types [2]:

1. *Symmetric Key Cryptosystems:* It is the oldest branch in the field of cryptology, and is still one of the most important ones today. Symmetric cryptosystems rely on a shared secret between communicating parties. In symmetric key encryption which is known as secret-key encryption, a single is used for both encryption and decryption. This key is kept secret by both the encryption party and decryption party [4, 5].

Symmetric ciphers usually fall into one of two categories: block cipher or stream ciphers [6].

2. *Asymmetric Key Cryptosystems:* It uses one key for encryption and another for decryption. The encryption key known as public key is intelligible and can be distributed for all parities, while the decryption key known as private key is intelligible only to the recipient. Each user creates a pair of keys; if one is used for encryption then the other is used for decryption [6, 7].

C. Classification According to Percentage of Encrypted Data

With respect to the amount of encrypted data, the encryption can be divided into full encryption and partial encryption (also called selective encryption), according to the percentage of the data encrypted [2].

III. MATHEMATICAL PRELIMINARIES

This section introduces the mathematical principles behind building the proposed cryptosystem.

A. Interpolation

Frequently you need to estimate intermediate values between precise data points. The most common method used for this purpose is polynomial interpolation. For $n + 1$ data points, there is one and only one polynomial of order n that passes through all the points. For example, there is only one straight line (that is, a first-order polynomial) that connects two points Figure (1(a)). Similarly, only one parabola connects a set of three points Figure (1(b)). Polynomial interpolation consists of determining the unique *nth*-order polynomial that fits $n + 1$ data points. This polynomial then provides a formula to intermediate values [8]. There are a variety of mathematical formats in which this polynomial can be expressed, such as Lagrange form.

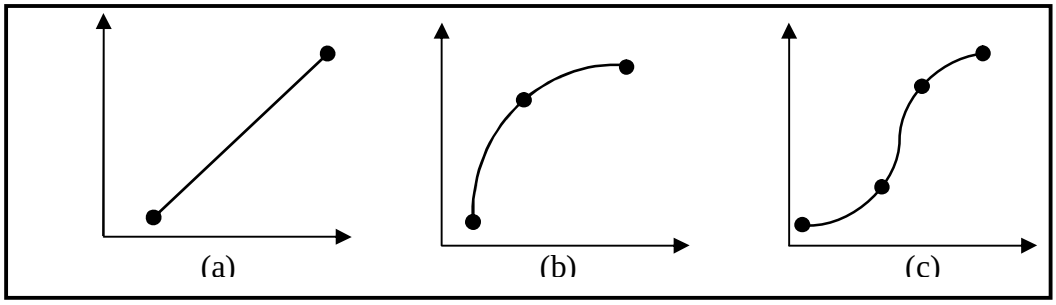


FIGURE 1: EXAMPLES OF INTERPOLATING POLYNOMIALS: (A) FIRST-ORDER (LINEAR) CONNECTING TWO POINTS, (B) SECOND ORDER (QUADRATIC OR PARABOLIC) CONNECTING THREE POINTS, AND (C) THIRD-ORDER (CUBIC) CONNECTING FOUR POINTS [8]

B. Lagrange Interpolation

It is always possible to construct a unique polynomial of degree n that passes through $n + 1$ distinct data points. One means of obtaining this polynomial is the formula of Lagrange. The general form of Lagrange interpolating polynomial is [9]:

$$P_n(x) = \sum_{i=0}^{n} L_i(x) \times f(x_i) \quad (1)$$

where the subscript *n* denotes the degree of the polynomial and

$$L_i(x) = \frac{x-x_0}{x_i-x_0} \bullet \frac{x-x_1}{x_i-x_1} \cdots \frac{x-x_{i-1}}{x_i-x_{i-1}} \bullet \frac{x-x_{i+1}}{x_i-x_{i+1}} \cdots \frac{x-x_n}{x_i-x_n} \quad (2)$$

$$= \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x-x_j}{x_i-x_j}, \quad i = 0, 1, \ldots\ldots\ldots, n$$

are called the cardinal functions.

Written explicitly:

$$P_n(x) = \frac{x-x_1}{x_0-x_1} \bullet \frac{x-x_2}{x_0-x_2} \cdots \frac{x-x_n}{x_0-x_n} \times y_0 + \frac{x-x_0}{x_1-x_0} \bullet \frac{x-x_2}{x_1-x_2} \cdots \frac{x-x_n}{x_1-x_n} \times y_1 + \cdots$$

$$+ \frac{x-x_0}{x_n-x_0} \bullet \frac{x-x_1}{x_n-x_1} \cdots \frac{x-x_{n-1}}{x_n-x_{n-1}} \times y_n \quad (3)$$

For example, if $n = 1$, the interpolation is the straight line (as shown in Figure (1(a))

$$P_1(x) = y_0 L_0(x) + y_1 L_1(x)$$

, where $L_0(x) = \frac{x-x_1}{x_0-x_1}$ $L_1(x) = \frac{x-x_0}{x_1-x_0}$

Example 1

By Lagrange formula, find the value of *f(3)* and *f(5)* from the following table.

X	0	1	2	4
F(X)	1	1	2	5

Solution:

$$f(3) = \sum_{j=0}^{3} f(x) \prod_{\substack{i=0 \\ i \neq j}}^{3} \frac{(3-x_i)}{(x_j-x_i)} = f(x_0)\frac{(3-x_1)(3-x_2)(3-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} + f(x_1)\frac{(3-x_0)(3-x_2)(3-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}$$

$$+ f(x_2)\frac{(3-x_0)(3-x_1)(3-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} + f(x3)\frac{(3-x_0)(3-x_1)(3-x2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}$$

f(3) = 3.5 by the same way for *f(5) = 6* [10].

C. Least Squares Method

A more appropriate strategy for such cases is to derive an approximating function that fits the shape or general trend of the data without necessarily matching the individual points [8]. Figure (2) illustrates how a straight line can be used to generally characterize trend of the data without

passing through any particular point. A technique for accomplishing this objective is called least squares method. The simplest example of a least-squares approximation is fitting a straight line (also called liner regression) to a set of paired observations: (x_0, y_0). (x_1, y_1), ..., (x_n, y_n). The mathematical expression for the straight line is [9, 8]:

$$y(x) = a + b\,x \tag{4}$$

where a, b are constant and can be obtained as follows:

$$a = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x^2{}_i - (\sum x_i)^2} \tag{5}$$

$$b = \bar{y} - a\bar{x} \tag{6}$$

where $\bar{y}$ and $\bar{x}$ are the means of y and x, respectively.

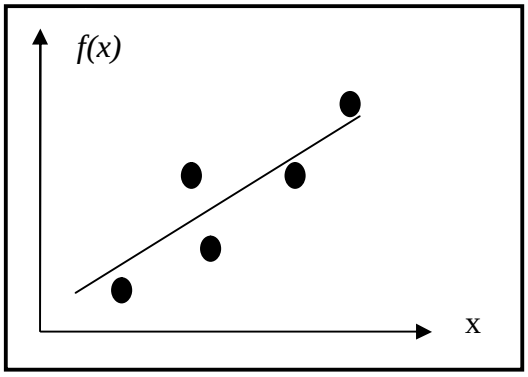


FIGURE 2: FITTING A STRAIGHT LINE THROUGH FIVE DATA POINTS BY USING LEAST SQUARES METHOD [8]

Example 2

Fit a straight line to the x and y values in the first two columns of the following table.

X_i	1	2	3	4	5	6	7
Y_i	0.5	2.5	2.0	4.0	3.5	6.0	5.5

The following quantities can be computed:

n=7, $\sum x_i y_i = 119.5$, $\sum x_i^2 = 140$

$\sum x_i = 28$, $\bar{x} = 28/7 = 4$

$\sum y_i = 24$, $\bar{y} = 24/7 = 3.428571$

$a = (7(119.5) - 28(24)) / (7(140) - (28)^2) = 0.8392857$

$b = 3.428571 - 0.8392857(4) = 0.07142857$

Therefore, the least squares fit is $y = 0.8392857 + 0.07142857x$ [8].

D. Discrete Wavelet Transform

Wavelet transform partitions a signal into a set of functions called wavelets. Wavelets are obtained from a single prototype wavelet called mother wavelet by dilations and shifting. Wavelet can represent a signal in time-frequency domain. Analyzing a signal with this kind of representation gives more information about the when and where of different frequency components [11].

IV. THE PROPOSED IMAGE CRYPTOSYSTEM

The proposed image cryptosystem (**XLLS**) is a symmetric stream cipher. This means that it uses the same key for both encryption and decryption processes, and encrypts the image pixel by pixel. The plain-image is given at its input. The **XLLS** cryptosystem consists of two main parts:

A. The Encryption/Decryption Algorithm

The encryption algorithm is composed of two main stages: the diffusion stage and the substitution stage. For the decryption algorithm, it is simply the reverse of the encryption algorithm. The general architecture of the proposed cryptosystem is shown in Figure (3).

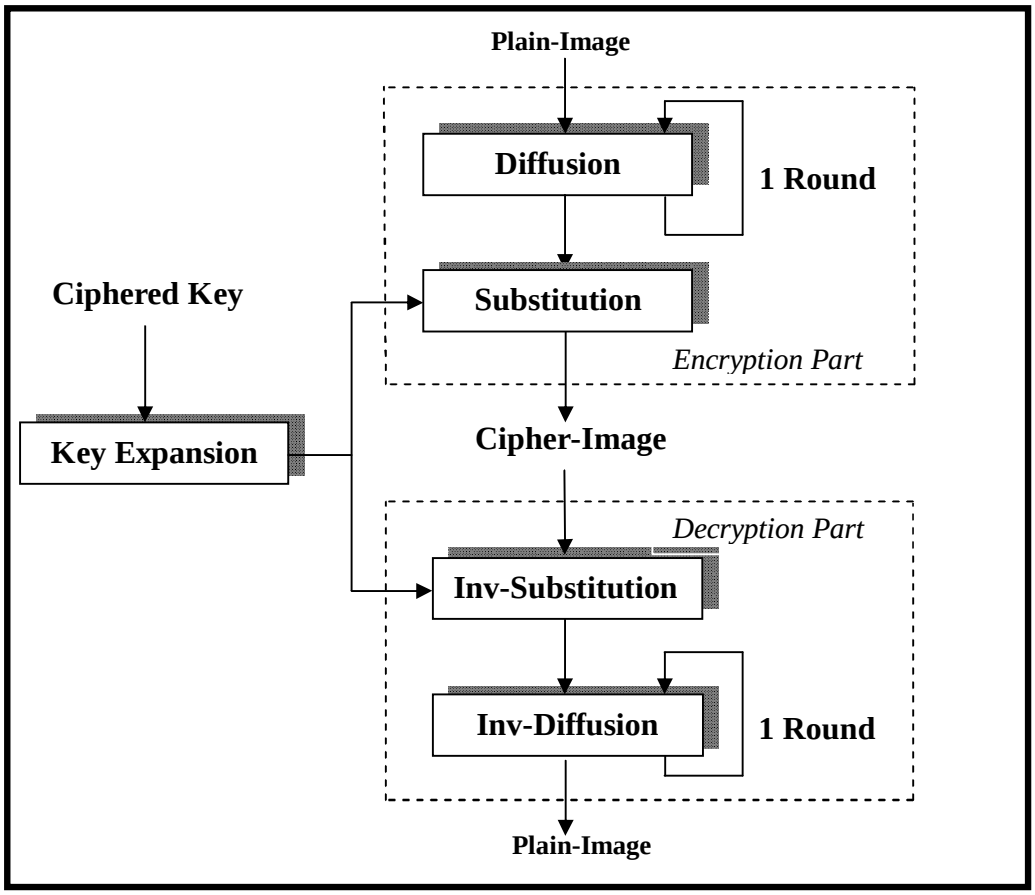


FIGURE3: THE GENERAL ARCHITECTURE OF THE PROPOSED CRYPTOSYSTEM

1. *The Diffusion Stage*: A slight change in the original image is reflected hopefully in all pixels of the cipher-image. A new mechanism of diffusion process is proposed in this paper. The proposed diffusion steps are as follows:
   - Generate an initial seed for the round as follows: One, take the value of last pixel in the current image, rotate left the bits of the value one position, and then save the result to the variable '***R***'. And two, add the value of variable '***R***' to the number **21** and the value of last pixel in current image, and then save the result to the variable '***S***'.
   - XOR each pixel (except the first one) with previous output (i.e., the current pixel is XORed with previously diffused pixel). For the first pixel, it's simply XORed with a variable '***S***'. This process continues until all pixels in the image are XORed, as in following formulas:

$$D_0 = P_0 \text{ XOR } S$$

$$D_i = P_i \text{ XOR } D_{i-1}, \quad i = 1, 2, \ldots\ldots, n \times m - 1 \qquad (7)$$

   where P_0 is the value of first pixel in the image and D_0 is the diffused value of P_0. P_i is the value of current pixel in the image. D_{i-1} is the value of previously diffused pixel of image and D_i is the value of current diffused pixel of image.
   - Repeat the above steps only for one time to ensure that each pixel in the image is affected by all other pixels.
2. *Substitution Stage:* This stage is composed of two encryption processes: Lagrange Process (LP) and Least Squares Process (LSP).

In the first encryption process, the proposed cryptosystem uses for encryption a first-order Lagrange interpolating polynomial. The Lagrange's form in cryptosystem is formulated as:

$$C_i = Y_o \times \frac{D_i - X_1}{X_0 - X_1} + Y_1 \times \frac{D_i - X_0}{X_1 - X_0} \qquad (8)$$

where the values of X_0 *and* Y_0 of the first point, and X_1 *and* Y_1 of the second point are obtained from key expansion generator, with $X_0 \neq X_1$. D_i is the value of current pixel and C_i is the ciphered pixel of D_i.

For decryption process, the Lagrange's equation is represented as follow:

$$D_i = X_o \times \frac{C_i - Y_1}{Y_0 - Y_1} + X_1 \times \frac{C_i - Y_0}{Y_1 - Y_0} \qquad (9)$$

On the other hand, in the second encryption process, the proposed cryptosystem uses least squares method to fit a straight line to a set of data by directly generating the values of parameters (*A* and *B*). A linear equation (i.e., least squares line) in **XLLS** cryptosystem is described by:

$$C_{ci} = A + C_i \times B \qquad (10)$$

where (*A* and *B*) parameters are obtained from key expansion generator. C_i is the current pixel and C_{ci} is the ciphered pixel of C_i.

For decryption process, the equation is represented as follow:

$$C_i = \frac{C_{ci} - A}{B} \tag{11}$$

B. Ciphered Key

The **XLLS** cryptosystem introduces two different approaches of initial key:

1. *A Sequence of hexadecimal characters:* In this approach, the proposed cryptosystem uses a key (k_0) whose length is of 192 bits (24 bytes) in hexadecimal characters as its input and expands it by using AES-192 key expansion algorithm. The expansion of ciphered key is based on the number of pixels in the plain-image. The AES key expansion algorithm may be repeated for a number of times until all pixels in the image are ciphered.
2. *Key-Image:* Usually in traditional cryptosystems, a ciphered key is a character or a sequence of characters. The proposed cryptosystem aims to use an image as a ciphered key. The direct encryption of an image by using any other image is not always successful. The reasons behind that are:
    - High correlation between pixels of the most images. This means that the key-image consists of values which are close to each other (this is a weak key).
    - Some images consist of only one color, such as a full black image. This means that the key-image consists of only one value (this is a very weak key).
    - You need to be very careful in choosing an image as a key (the key-image should have as many colors and edges as possible).
    - If the attacker knows a part of key-image, he sometimes simply can deduce the rest of key-image.

To avoid those problems, a new mechanism of key expansion algorithm is proposed in this paper, called **CBI**. It uses an image as its input. The size of key-image must be equal to the plain-image. The procedure of **CBI** is summarized as follows:

- Load the key-image into $\boldsymbol{Kmg_i}$ array, $\boldsymbol{Kmg} = \{\boldsymbol{P_0}, \boldsymbol{P_1}, \ldots, \boldsymbol{P_{n\times m-1}}\}$.
- The three different layers of pixels corresponding to red, green and blue channels of key-image are separated out and transferred to three independent arrays ($\boldsymbol{Red_i}$, $\boldsymbol{Green_i}$ and $\boldsymbol{Blue_i}$)**,** respecitvely.
- Produce the intensity value (gray-scale) of the key-image, and save the result into an array called $\boldsymbol{Gray_i}$ by using the following formula:

$$\boldsymbol{Gray_i} = 0.299 * \boldsymbol{Red_i} + 0.587 * \boldsymbol{Green_i} + 0.114 * \boldsymbol{Blue_i} \tag{12}$$

- Input each color array ($\boldsymbol{Red_i}$, $\boldsymbol{Green_i}$, $\boldsymbol{Blue_i}$ and $\boldsymbol{Gray_i}$) independently to XOR diffusion process, and save the results into $\boldsymbol{DRed_i}$, $\boldsymbol{DGreen_i}$, $\boldsymbol{DBlue_i}$ and $\boldsymbol{DGray_i}$ arrays, respectively.

- Take the values of four predetermined locations in the gray-level diffused array and set them to the variables: ***Vred***, ***Vgreen***, ***Vblue*** and ***Vgray***, as following:

  Vred = DGray*(20)*, ***Vgreen = DGray****(10)*, ***Vblue = DGray****(27)* and ***Vgray = DGray****(16)*

- Input each diffused array (***DRed****i*, ***DGreen****i*, ***DBlue****i* and ***DGray****i*) to a new process called Simple Substitution Process (**SSP**). The **SSP** processes each array independently pixel by pixel. Each pixel passes through all **SSP** steps before moving to the next one. The **SSP** steps for the four diffused arrays are as follows:
  - Shift left bits of the current pixel in each array by one position.
  - XOR the current pixel in the arrays with its eighth bit and save the resulting into the same pixel.
  - Input the current pixel of each array into its corresponding equation, as follows:

$$KR_i = DRed_i \text{ XOR } (Vred * i) \bmod 256, \text{ for red diffused array}$$
$$KG_i = DGreen_i \text{ XOR } (Vgreen * i) \bmod 256, \text{ for green diffused array}$$
$$KB_i = DBlue_i \text{ XOR } (Vblue * i) \bmod 256, \text{ for blue diffused array}$$
$$KGR_i = DGray_i \text{ XOR } (Vgray * i) \bmod 256, \text{ for gray diffused array} \quad (13)$$

    where $i = 0 \text{ to } n \times m - 1$

  - XOR the pixel (except the first pixel) with previous modified element, as follows:

$$KR_i = KR_i \text{ XOR } KR_{i-1}$$
$$KG_i = KG_i \text{ XOR } KG_{i-1}$$
$$KB_i = KB_i \text{ XOR } KB_{i-1}$$
$$KGR_i = KGR_i \text{ XOR } KGR_{i-1} \quad (14)$$

  - Rotate left bits of each variable (***Vred***, ***Vgreen***, ***Vblue*** and ***Vgray***) independently by one position.
  - XOR the variables (***Vred***, ***Vgreen***, ***Vblue*** and ***Vgray***) with (2, 3, 8 and 1) values, respectively, and save the resulting into the same variables.
  - Repeat previous steps to the last pixel.

The values of variables (X_0, Y_0, X_1 and Y_1) of Lagrange process in substitution stage will be taken from ***KR****i*, ***KG****i*, ***KB****i* and ***KGR****i* arrays, respectively. On the other hand, the values of variables (A and B) of least squares process will be taken from ***KB****i* and ***KGR****i* arrays, respectively.

III. The Proposed Encryption Techniques

The **XLLS** cryptosystem has been used in two different encryption approaches, and the users have option to choose any one of them to encrypt the plain-image. These approaches are:

A. Full Encryption

In this approach, all pixels of the plain-image are encrypted by using the **XLLS** cryptosystem. The result is the encrypted image (cipher-image).

B. **Partial Encryption**

In this approach, only a part of the original image will be encrypted. The encryption scheme presented here is based on the DWT and **XLLS** cryptosystem. This technique has benefits in encryption:

- The average time used in encryption/decryption process is very short.
- Low storage requirements for the resulting image.

In the DWT method, the original image of size ($R \times S$) is decomposed into four sub-bands (sub-images) each of size ($R/2 \times S/2$) containing information from different frequency components. The LL sub-band contains an approximation of the original image while the other sub-bands contain the missing details. The LL sub-image will be encrypted because it holds most of the image's information by using the **XLLS** cryptosystem. For all other sub-bands, they will be deleted.

V. IMPLEMENTATION AND EXPERIMENTAL RESULTS

In this section, the **XLLS** cryptosystem is implemented and its experimental results are investigated to demonstrate both the efficiency and the security of the scheme. The **XLLS** cryptosystem was implemented in Visual Basic.NET 2008 where the implementation was done on Intel(R) Core (TM) i3 processor with 2.00 GB RAM under Windows 7 Ultimate SP1.

FIGURE 4: THE COLOR KEY–IMAGE "LENA"

A. Information Entropy Analysis

Entropy of a source gives idea about self information i.e., information provided by a random process about itself. The concept of entropy is very important for analyzing an encryption scheme [12]. Information entropy is the most important feature of randomness. Let S be the information source, and the formula for calculating information entropy is [13]:

$$H(m) = -\sum_{i=0}^{2^n-1} P(S_i) log_2[P(S_i)] \quad (15)$$

where $P(S_i)$ denotes the probability of symbol S_i, 2^n is the total state of the information source. For a true random source emitting 2^n symbols, the entropy should be n. Take a 256 gray-scale image for example, and the pixel data have 2^8 possible values, so the ideal entropy of a 256 gray-scale image must be 8.

The entropy values for the four test original images and their corresponding ciphered forms that are illustrated in Figure (5) are calculated and presented in first column of Table (1). It's clear that the entropy values of cipher-images are very close to the theoretical value 8 (which is the best) contrary to their original forms. This means that the proposed cryptosystem is secure upon entropy attack.

B. Correlations of Two Adjacent Pixels

There exists a high correlation between pixels of an image which is called intrinsic feature. Thus, a secure encryption scheme should remove it to improve resistance against statistical analysis. To test the correlation between two-adjacent pixels in plain-image and cipher-image, all N pairs of two-adjacent (in vertical, horizontal, and diagonal direction) pixels from plain-image and cipher-image are selected and the correlation coefficients are calculated by using the following formulas:

$$E(X) = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{16}$$

$$D(X) = \frac{1}{N}\sum_{i=1}^{N} [X_i - E(X)]^2 \tag{17}$$

$$Conv = \frac{1}{N}\sum_{i=1}^{N} [X_i - E(X)][Y_i - E(Y)] \tag{18}$$

$$\gamma_{xy} = \frac{Conv(x,y)}{\sqrt{D(X)}\sqrt{D(Y)}} \tag{19}$$

$$With\ D(X) \neq 0\ and\ D(Y) \neq 0$$

where x and y are gray-scale values of two-adjacent pixels in the image, γ_{xy} is the correlation coefficient of two adjacent pixels x and y [13].

The second column of Table (1) gives the correlation coefficient values of adjacent pixels in the horizontal, vertical, and diagonal directions of the original images and their corresponding ciphered forms. It's clear that the coefficient correlation values of the original images are very high (close to one) contrary to those observed for the encrypted images which are zero. This security analysis proves that the proposed cryptosystem achieves zero coefficient correlation between adjacent pixels, which is of high-level security.

C. Correlations between Original and Ciphered Images

The correlation between various pairs of plain and cipher-images is analyzed too by computing the 2D correlation coefficients (C.C) between original and encrypted images. The C.C is calculated as follows:

$$C.C = \frac{\sum_{i=1}^{M1}\sum_{j=1}^{M2}(A_{ij}-\overline{A})(B_{ij}-\overline{B})}{\sqrt{(\sum_{i=1}^{M1}\sum_{j=1}^{M2}(A_{ij}-\overline{A})^2)(\sum_{i=1}^{M1}\sum_{j=1}^{M2}(B_{ij}-\overline{B})^2)}} \quad (20)$$

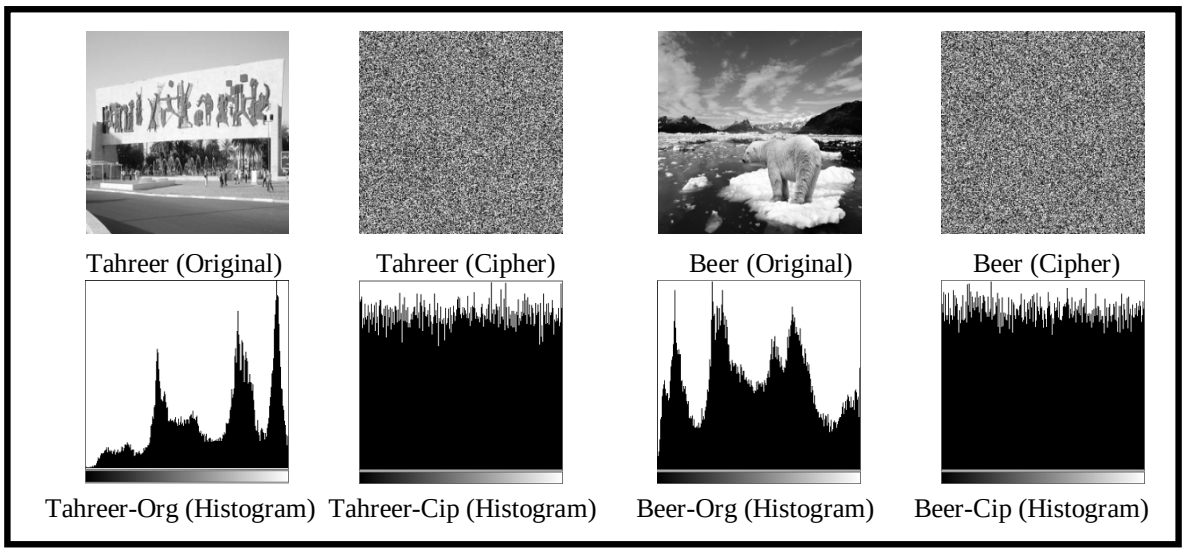
$$\text{With } \overline{A}=\frac{1}{M1\times M2}\sum_{i=1}^{M1}\sum_{j=1}^{M2}A_{ij} \text{ and } \overline{B}=\frac{1}{M1\times M2}\sum_{i=1}^{M1}\sum_{j=1}^{M2}B_{ij}$$

where A represents the plain-image, B represents the cipher-image. $\overline{A}$ and $\overline{B}$ are the mean values of the elements of matrices A and B, respectively. M1 and M2 are the height and width of the plain/cipher-image, respectively [13].

The third column of Table (1) shows the 2D correlation coefficients (C.C) between the original images and their corresponding ciphered forms. It is clear that no correlation (zero correlation) exists between each original image and its corresponding ciphered image. Consequently, the proposed cryptosystem successfully passes this test.

D. Cipher-Image Histogram Analysis

Figure (5) shows the four original images, corresponding cipher forms and their histograms. It's clear that the histograms of cipher-images are approximately uniformly distributed and are significantly different from their corresponding original images. Hence, the cryptosystem protects the plain-image information against statistical attack.



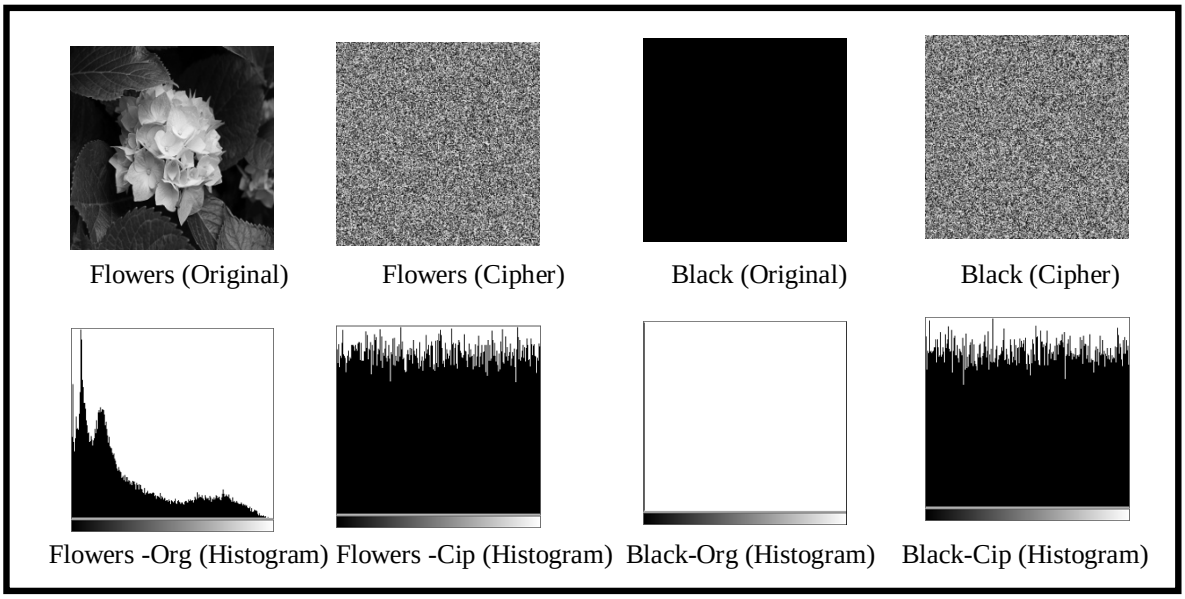


FIGURE 5: THE ORIGINAL IMAGES, CIPHERED IMAGES AND THEIR HISTOGRAMS

TABLE 1: THE ENTROPY, CORRELATIONS AND C.C VALUES FOR THE TEST IMAGES

Images		Entropy	Correlation			C.C
			Horizontal	Vertical	Diagonal	
Tahreer	Plain	7.5934	0.9547	0.9052	0.8812	-0.0015
	Cipher	7.9964	-0.0013	0.0018	-0.0001	
Beer	Plain	7.8862	0.9367	0.9672	0.9229	-0.0001
	Cipher	7.9969	-0.0027	0.0048	0.0018	
Flowers	Plain	7.4364	0.9551	0.9447	0.9619	-0.0020
	Cipher	7.9968	-0.0055	0.0057	-0.0031	
Black	Plain	0	1	1	1	0
	Cipher	7.9965	-0.0003	0.0067	-0.0027	

E. Key Space Analysis

The key space should be large enough to make brute force attacks infeasible. As mentioned, the proposed cryptosystem introduces two different approaches of initial key.

In the first approach, the proposed cryptosystem uses a sequence of hex characters as a key of length 192 bits. This means that it has 2^{192} ($\approx 6.2771\times10^{57}$) different combinations of secret key.

In the second approach, the proposed cryptosystem uses an image as a key; the size of key is variable and dependent on the size of the original image. If the original image size increases, the size of key space increases accordingly. This means that the proposed cryptosystem has:

$2^{w\times h\times 24}$ of different combinations of secret key.

where w and h are the width and the height of key-image, respectively; 24 is a number of bits to represent each pixel.

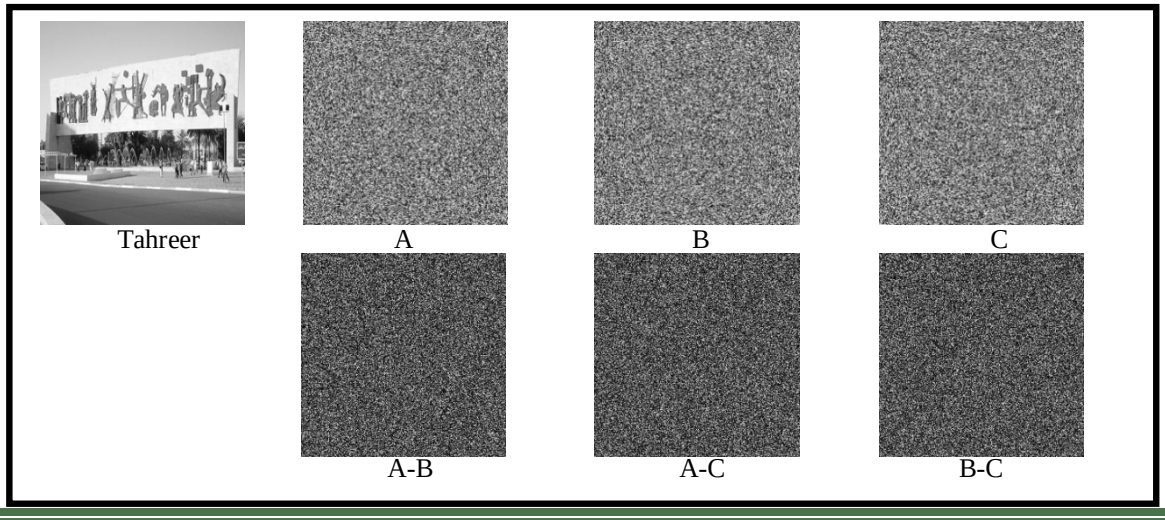
It's clear from above; the proposed cryptosystem has such a large key space that provides a sufficient security against all kinds of brute force attacks.

F. Key Sensitivity Test

Encryption algorithms should be high sensitive to ciphered key: this means that a slight change in the key should lead to a significant change in the encrypted or decrypted image. To illustrate the key sensitive of the proposed cryptosystem upon using hex key as its initial key, the following steps have been performed:

- An original image is encrypted by using the secret key*"A9C39 01376581987BB2657A3501658723B657EA2BFFEC891 "* and the resulting image is referred to as encrypted image A.
- The same original image is encrypted by making a slight modification in the secret key (i.e., the most significant bit is changed in the secret key) *"**2**9C3901376581987BB2657A350165 8723B657EA2BFFEC891"* and the resulting image is referred to as encrypted image B.
- The same original image is encrypted by making a slight modification in the secret key (i.e., the least significant bit is changed in the secret key) *"A9C3901376581987BB2657A3501 658723B657EA2BFFEC89**0**"* and the resulting image is referred to as encrypted image C.
- Finally, the three encrypted images A, B and C are compared.

Figure (6) shows the original images of (Tahreer and Beer), in addition to the three encrypted images and the three difference images (A-B, A-C and B-C) of each original form. The 2D correlation coefficients and pixel difference between the corresponding pixels of the three encrypted images A, B and C of each original image are calculated and presented in Table (2). It's clear that there is no correlation (zero correlation) among three encrypted images even though they have been produced by using slightly different secret keys.



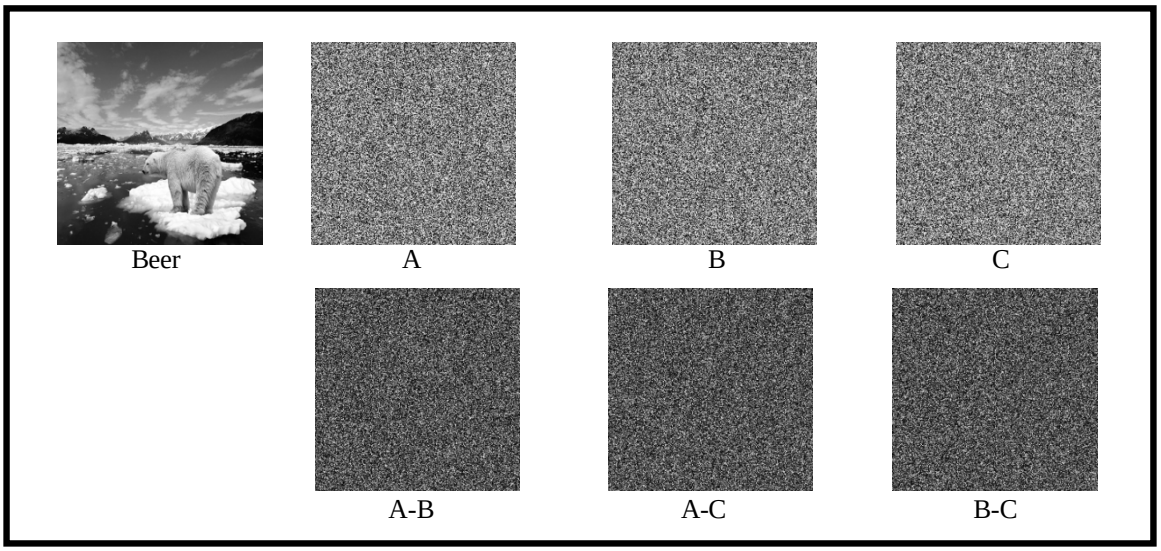


FIGURE 6: THE KEY SENSITIVITY TEST BY ENCRYPTING THE ORIGINAL IMAGES BY SEVERAL SLIGHTLY DIFFERENT KEYS

TABLE 2: THE 2D CORRELATION COEFFICIENTS (C.C) AND PIXEL DIFFERENCE BETWEEN THE CORRESPONDING PIXELS OF THE THREE DIFFERENT ENCRYPTED IMAGES RESULTING BY USING SLIGHTLY DIFFERENT SECRET KEYS

Images	Encrypted image 1	Encrypted image 2	C.C	Pixel difference
Tahreer	Image A	Image B	-0.0018	99.62%
	Image A	Image C	-0.0030	99.58%
	Image B	Image C	-0.0019	99.62%
Beer	Image A	Image B	0.0032	99.54%
	Image A	Image C	0.0007	99.63%
	Image B	Image C	-0.0025	99.59%

Another test has been performed to illustrate the key sensitivity of the proposed cryptosystem, but for image decryption process. When a key is used to encrypt an image, another slightly modified key (differs in only one bit) is used to decrypt the ciphered image, the decryption also completely fails. Figure (7) shows the original images of (Tahreer and Beer) encrypted by*"A9C3901376581987BB2657A3 501658723B657EA2BFFEC891"* (image A) are not correctly decrypted by using the key *"**2**9C3901376581987BB2657A3501658723 B657E A2BFFEC891"* (image B) or by the key*"A9C3901376581987BB2657A3501658723 B657EA2BFFEC89**0**"* (image C), although there is only one bit difference between the encryption and decryption keys.

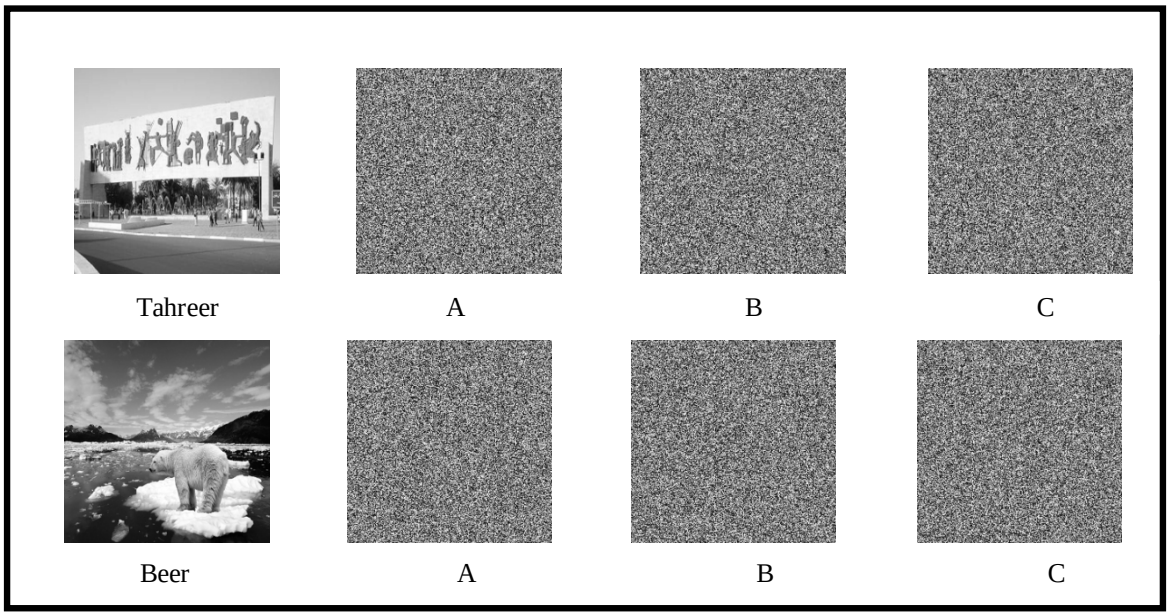
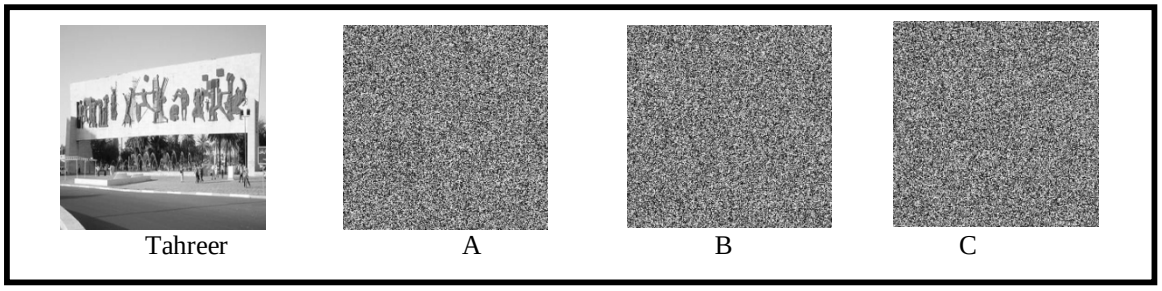


FIGURE 7: KEY SENSITIVITY TEST FOR DECRYPTION PROCESS BY USING SLIGHTLY DIFFERENT KEYS FROM THE ENCRYPTION KEY

To illustrate the key sensitive of the proposed cryptosystem when using a key-image as its initial key, the following steps have been performed:

- An original image is encrypted by using the secret key-image of Lena and the resulting image is referred to as encrypted image A. The key-image of Lena is shown in Figure (4).
- The same original image is encrypted by making a slight modification in the secret key-image (i.e., the center pixel value of key-image of Lena is 63 which is then changed to be 0). The resulting image is referred to as encrypted image B.
- The same original image is encrypted by making a slight modification in the secret key (i.e., the first pixel value in the upper left corner of key-image of Lena is 127 which is then changed to be 0). The resulting image is referred to as encrypted image C.
- Finally, the three encrypted images A, B and C are compared.

Figure (8) shows the original images of (Tahreer and Beer), in addition to the three encrypted images and the three difference images (A-B, A-C and B-C) of each original form. The correlation coefficients and pixel difference between the corresponding pixels of the three encrypted images A, B and C of each original image are calculated and presented in Table (3). It's clear that there is no correlation (zero correlation) among three encrypted images even though they have been produced by using slightly different secret key-images.



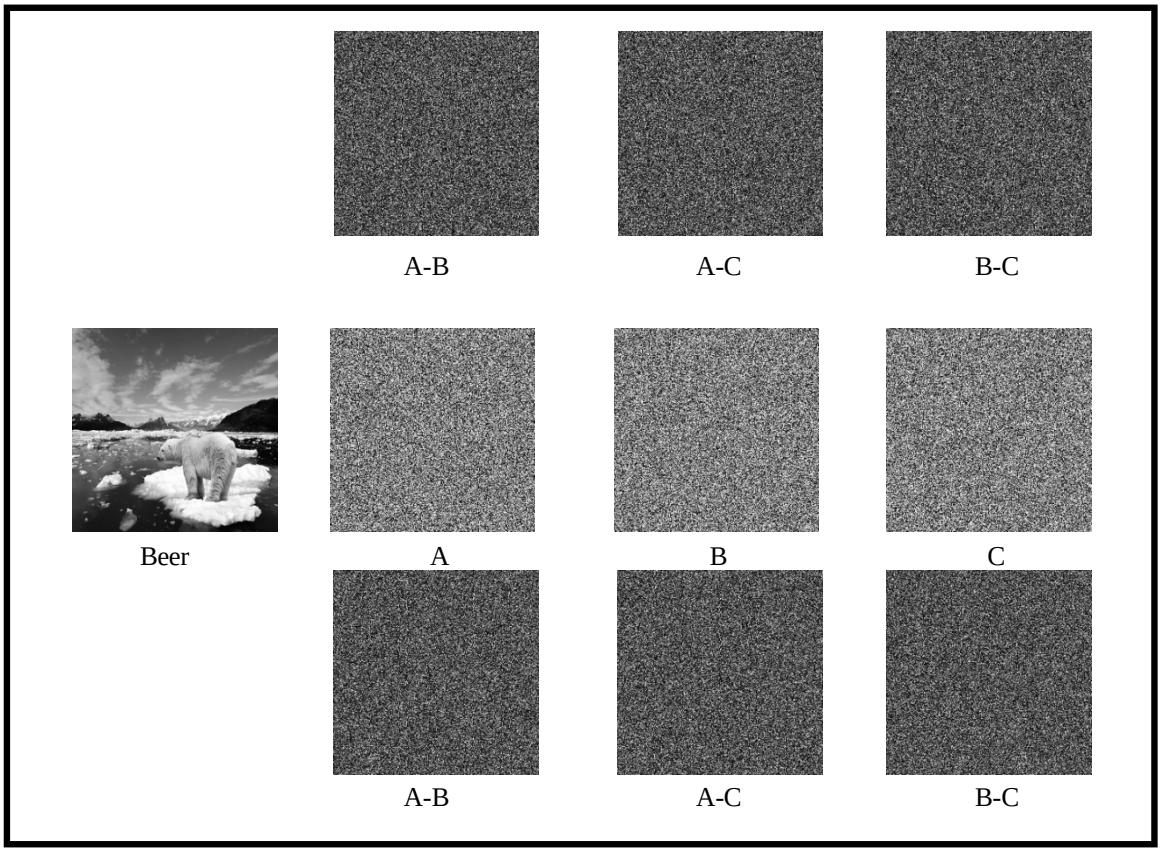


FIGURE 8: THE KEY SENSITIVITY TEST BY ENCRYPTING THE ORIGINAL IMAGES BY SEVERAL SLIGHTLY DIFFERENT KEYS

TABLE 3: THE CORRELATION COEFFICIENTS (C.C) AND PIXEL DIFFERENCE BETWEEN THE CORRESPONDING PIXELS OF THE THREE DIFFERENT ENCRYPTED IMAGES RESULTING BY USING SLIGHTLY DIFFERENT SECRET KEY-IMAGES

Images	Encrypted image 1	Encrypted image 2	C.C	Pixel difference
Tahreer	Image A	Image B	0.0055	99.58%
	Image A	Image C	-0.0060	99.55%
	Image B	Image C	-0.0051	99.58%
Beer	Image A	Image B	0.0046	99.55%
	Image A	Image C	-0.0062	99.58%
	Image B	Image C	-0.0005	99.59%

When a key-image is used to encrypt an image, another slightly modified key-image (differs in only one bit) is used to decrypt the ciphered image, the decryption also completely fails. Figure (9) shows the original images of (Tahreer and Beer) encrypted by the key-image of Lena (image A) are not correctly decrypted by making a slight modification in the center pixel value of the key-image (image B) or by making a slight modification in the first pixel that is located in the upper left corner of the key-image (image C), although there is only one bit difference between the

encryption and decryption keys. This proves that the proposed cryptosystem has high sensitivity to encryption key.

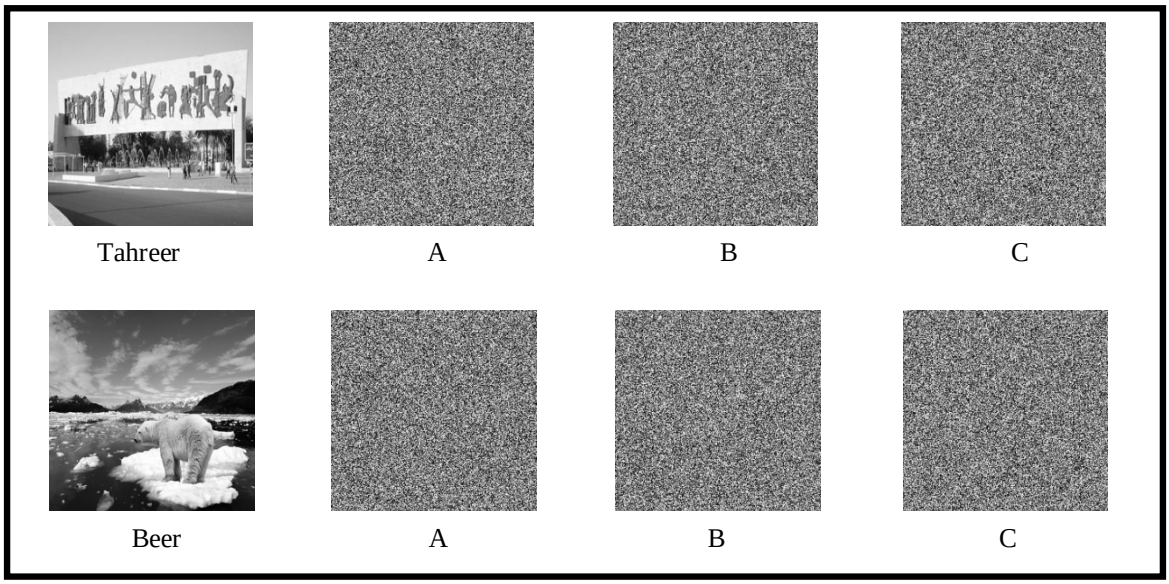


FIGURE 9: KEY SENSITIVITY TEST FOR DECRYPTION PROCESS BY USING SLIGHTLY DIFFERENT KEYS FROM THE ENCRYPTION KEY-IMAGE

G. Differential Analysis

A desirable property for the proposed cryptosystem is the high sensitivity to small changes in the plain-image (single bit change in the plain-image). In general, an opponent may make a slight change, such as modifying only one pixel of the original image, and then observing the change of the result. In this way, he may be able to find out a meaningful relationship between the plain-image and the cipher-image. If a minor change in the plain-image can cause a significant change in the cipher- image, then this differential attack would become very inefficient and practically useless. To test the influence of a one-pixel change on the whole image encrypted by the proposed cryptosystem, two common measures may be used: the Number-of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI). Let two ciphered images, whose corresponding plain-images have only one-pixel difference, be denoted by C_1 and C_2. Label the gray-scale values of the pixels at grid (i , j) in C_1 and C_2 by $C_1(i , j)$ and $C_2(i , j)$, respectively. Define a bipolar array (D) with the same size as the images C_1 and C_2 [14]. The D(i, j) represents the difference between $C_1(i , j)$ and $C_2(i , j)$. If $C_1(i , j) = C_2(i , j)$ then D(i, j)=0, otherwise D(i, j)=1[13]. The NPCR is defined as:

$$\text{NPCR} = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \qquad (21)$$

where W and H are the width and height of C_1 or C_2. The NPCR measures the percentage of the number of different pixels to the total number of pixels between these two images. The UACI is defined as:

$$\text{UACI} = \frac{1}{W \times H}\left[\sum_{i,j}\frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\% \qquad (22)$$

The NPCR and UACI values for the test images are calculated and presented in Table (4). It's clear from the table that the **XLLS** cryptosystem has high sensitivity to small changes in the original image.

TABLE 4: THE NPCR AND UACI FOR THE TEST IMAGES

Images	NPCR	UACI
Tahreer	99.32%	31.09%
Beer	99.47%	32.66%
Flowers	99.32%	31.20%
Black	99.60%	33.54%

H. Time Analysis

In this section, the time analysis of the proposed cryptosystem has been investigated. Table (5) shows the encryption time for the four original images. Both the encryption approaches (full encryption and selective encryption) of the proposed cryptosystem have been applied to each test image.

TABLE 5: THE ENCRYPTION TIME ANALYSIS FOR THE PROPOSED CRYPTOSYSTEM

Images	Image Size	Encryption Time in Second (*s*)	
		Full Encryption	Selective Encryption
Tahreer	256×256	0.40	0.14
Beer	256×256	0.42	0.12
Flowers	256×256	0.34	0.07
Black	256×256	0.37	0.07

VI. CONCLUSIONS

A new image encryption scheme is presented in this paper. It provides both high security and high performance in encryption. Also, a new mechanism of diffusion process is identified depending on XOR operation. It makes the proposed cryptosystem very sensitive to small change in the original image. Hence the proposed cryptosystem resists successfully the differential attack.

The proposed cryptosystem can be used in two different encryption approaches: full encryption and selective encryption, and the user has option to choose any one of them to encrypt the required image. The proposed cryptosystem encrypts the original image by providing two different approaches of encrypted key: a sequence of hex characters and a key-image. It expands the hex key by using AES-192 expansion algorithm, while it processes and expands the key-image by identifying a new expansion algorithm for that purpose, called **CBI** expansion algorithm.

ACKNOWLEGOMENT

Firstly all my prayers be to (Allah), for the successive blessing and the success in my thesis. My greatest thanks are due to my supervisor Dr. Haider K. Hoomod for his guidance and supporting. Also, I would like to thank all my family members for their help in good and difficult times.

REFERNCES

[1] M, J. & S, M. (March 2012). A Survey on Various Encryption Techniques. International Journal of Soft Computing and Engineering (IJSCE), Vol. 2, Issue No. 1, pp. 429-432.

[2] El-Ashry, I. (2010). Digital Image Encryption. MS.c Thesis, Electronics and Electrical Communications Engineering Dept., Faculty of Electronic Engineering, Menofia University.

[3] Hung, K. (September 2007). A Study on Efficient Chaotic Image Encryption Schemes. MS.c Thesis, Electronic Engineering Dept., City University of Hong Kong.

[4] Jolfaei, A. & Mirghadri A. (September 2010). Survey: Image Encryption Using Salsa20. IJCSI International Journal of Computer Science Issues, Vol. 7, Issue No. 5, pp. 213-220.

[5] Mcgregor, J. (June 2005). Architectural Techniques for Enabling Secure Cryptographic Processing. PhD Thesis, Electrical Engineer Dept., Princeton University, New Jersey, United States.

[6] Ali, M. (2008). Scrambling and Encrypting Using Cipher Parameters Hopping, MS.c Thesis, Control and Computers Engineering Dept., University of Baghdad, Iraq.

[7] Younes, M. (2009). An Approach to Enhance Image Encryption Using Block Based Transformation Algorithm. PhD Thesis, the School of Computer Science, University Sains Malaysia.

[8] Chapra, S. & Canale, R. (2009). Numerical Methods for Engineers. Sixth Edition, McGraw-Hill College.

[9] Kiusalaas, J. (2005). Numerical Methods in Engineering with Python. First Edition, Cambridge University Press.

[10] Akif, O. (2012). Image Encryption Technique Using Lagrange Interpolation. Ibn Al-Haitham Journal for Pure and Applied Science, Vol. 25, No. 1.

[11] Abbasfard, M. (2009). Digital Image Watermarking Robustness: A Comparative Study. MS.c Thesis, Electrical Engineering Dept., Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, South Holland.

[12] Ahmad, J. & Ahmed, F. (August 2008). Efficiency Analysis and Security Evaluation of Image Encryption Schemes. International Journal of Video & Image Processing and Network Security, Vol. 12, No. 4, pp. 18-31.

[13] Congxu, Z. (1 January 2012). A Novel Image Encryption Scheme Based on Improved Hyperchaotic Sequences. Optics Communications, Vol. 285, Issue No. 1, pp. 29–37.

[14] Elashry, I., Farag Allah, O., Abbas, A., El-Rabaie, S. and Abd El-Samie, F.(July-September 2009). Homomorphic Image Encryption. Journal of Electronic Imaging, Vol. 18, No. 3, pp. 1-14.

AUTHORS' BIOGRAPHY

My name is Haider Kadim Hoomod, was born in Baghdad at 1974. I have B.Sc., M.Sc. and Ph.D. in Electrical Engineering (1997), Computer Security (2002) and Intelligent Network Security (2008), respectively. I am head of Computer Science Department in Education collage of Al-Mustansirya University. I have more 20 papers in several areas of computer science especially in data security. I supervised on 8 M.Sc. Thesis and 2 Ph.D. Dissertation.

My name is Mohammed Abud Al-Manuf Shreef, was born in Iraq / Baghdad at 1987, I have B.Sc. in Computer Engineering from the University of Al-Mustansirya, Iraq / Baghdad, 2009. I'm currently studying at the Institute of Informatics of Higher Studies in the Iraqi Commission for Computers and Informatics to obtain a Master's degree of Science in Software Engineering.