

# Hybrid method for ciphering colored image

Salman Abd Kadum, Ali Abdul Azeez Mohammad Baker

Computer Systems Techniques, Al Furat Al Wast University, Al Najaf Technical Institute, Al Najaf, Iraq.

## Email address:

salman\_abd\_2002@yahoo.com (S. A. Kadum), alia.qazzaz@uokufa.edu.iq (A. A. A. M. Baker)

## To cite this article:

Salman Abd Kadum, Ali Abdul Azeez Mohammad Baker. Hybrid method for Ciphering Colored Image. *Automation, Control and Intelligent Systems*. Special Issue: Artificial Nano Sensory System. Vol. 3, No. 2-1, 2015, pp. 22-34. doi: 10.11648/j.acis.s.2015030201.15

---

**Abstract:** Security has more and more importance in current era especially in the organizations where information is more critical and more effective, so there are many methods that will be used in keeping information secure and safety like steganography, cryptography, barcode, password, and biometrics. different techniques will be used to protect image data from unauthorized access, In this paper a hybrid method will be proposed for ciphering secure image by translating information from original RGB image to  $YC_bC_r$  format then using public key algorithm after normalized pixels values to insure that will be within public and prevent keys region after that the Modified Arnold Transform, Generalized Fibonacci Transform, and Fibonacci-Lucas Transform will be applied on each ciphered band from  $YC_bC_r$  image respectively.

**Keywords:** Cryptography, Ciphering, Deciphering, Color image, Public Key, Private Key, Arnold Transform, Fibonacci Transform, Fibonacci-Lucas Transform

---

## 1. Introduction

Cryptography is the science of hiding information, or is a fundamental tool for the protection of sensitive information. Encryption is a way of talking to someone while other people are listening, but such that the other people cannot understand what you are saying. It can also be used to protect data in storage. The goal of cryptography is to make data unreadable by unwanted persons. According to the key that will be used in ciphering algorithm, Cryptography algorithms are divided into two types, symmetric key and public-key (asymmetric) algorithms. Symmetric algorithms are used the same key in both ciphering and deciphering algorithms. While Public-key encryption algorithms used a pair of keys, the first key is used in encryption information that will be sent to a receiver

which is named (public) or the second key who owns the corresponding private key which is used in decryption the information. Image encryption techniques are used to overcome the problem of secure transmission for both images and text over the electronic media by using the suitable cryptographic algorithms.

### 1.1. Public Key Cryptography

Public key algorithm is a powerful method for ciphering information, and the strength of this algorithm depends on the strength of the public and private keys. As we know that every person who wants to send secret information by using unsecured channel must used a suitable way to protect this information, one of the most important protection method which is public key algorithm.

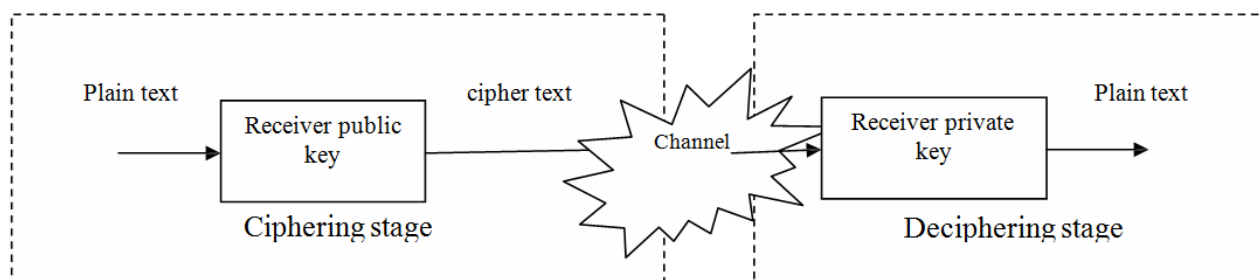


Figure (1). Public key encryption method.

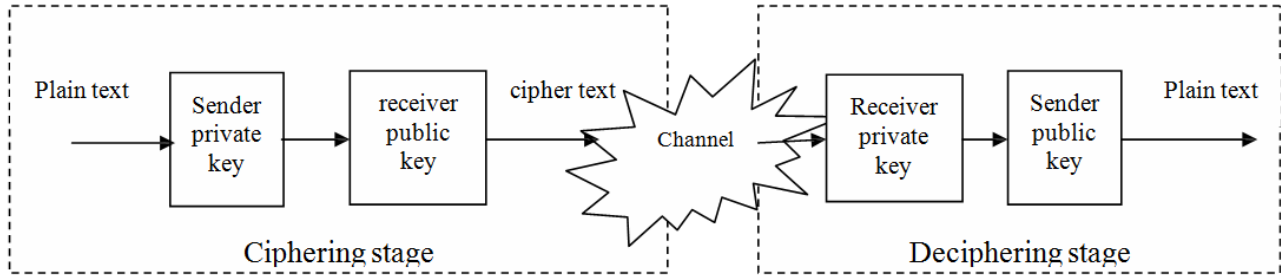


Figure (2). Signature encryption method.

Every person has two related keys (public and private), and if any person want to send secrete message to another person, he must used the public key of the received person to cipher the message and then send it by using any transform media, after that the received person must be deciphering the message to extract the information, and this operation done by using receiver private key as illustrated in figure (1), in some cases sender ciphering the message by using sender private key firstly then receiver public key and this state called the signature, so the receiver deciphering the message by using receiver private key firstly, then sender public key to extract information as illustrated in figure(2).

The two keys of any person are related and they created from primary numbers by using special method as follows:

- Selecting two big prime numbers (p, q), and big number (e) randomly.
- Calculating  $\theta(n) = (p-1) \times (q-1), n = p \times q$ .
- Calculating  $d = \frac{GCD((p-1), (q-1)) \times \theta(n) + 1}{e}$ .
- Constructing public key (e, n), and private key (d, n).

For example if person1 wants to construct his keys, firstly he chooses  $p = 23, q = 17, e = 15$ .

$$\theta(n) = (23-1) \times (17-1) = 352, n = 23 \times 17 = 391$$

$$d = \frac{GCD(22, 16) \times 352 + 1}{15} = \frac{705}{15} = 47$$

Then the public key of person1 is (15, 391), and the private key of person1 is (47, 391).

For ciphering any plain text by using public key algorithm the equation (1) will be used as follows:

$$C_{(number)} = (number)^{er} \mod n_r \quad (1)$$

Where  $(er, n_r)$  is the receiver public key.

For example if any person wants to send message (HI) to person1 with above keys then,

$$ASCII(H) = 72, ASCII(I) = 73$$

$$C(H) = (72)^{15} \mod 391 = 81$$

$$C(H) = (73)^{15} \mod 391 = 279$$

Therefore the person sends the message 81, 279 to person1.

For deciphering any cipher text by using public key algorithm the equation (2) will be used as follows:

$$P_{(number)} = (number)^{dr} \mod n_r, \quad (2)$$

Where  $(dr, n_r)$  is the receiver private key.

For the above cipher text that received by person1, he will extract plain text by applying equation (2) as follows:

$$P_{(81)} = (81)^{47} \mod 391 = 72 = H$$

$$P_{(279)} = (279)^{47} \mod 391 = 73 = I$$

Plain text is (HI)

## 1.2. Arnold Transform

It is a transform  $\Gamma: T^2 \rightarrow T^2$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \pmod{N} \quad (3)$$

Where  $x, y, x', y' \in \{0, 1, 2, \dots, N-1\}$ , and N is the size of an original squared image.

A new image will be produced when all points in the original image are ciphered by equation (1). The Arnold transform change the position of each pixel in original image but the value of this pixel stay with same original value, this transform is simple but powerful which is very much easy in spatial domain.

The security level of the encrypted image becomes danger when it is encrypted by using the basic Arnold transform as it depend on a single 2x2 array so, to enhance the security level of the encrypted image the basic Arnold transform is modified as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} k+1 & k \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \pmod{N} \quad (4)$$

Where  $x, y, x', y' \in \{0, 1, 2, \dots, N-1\}$ , N is the size of a digital squared image, and  $k = \{0, 1, 2, 3, \dots\}$

Unlike equation (1) where there is a single map, equation (4) provides a new map for each value of k so; the security level of the scrambled image against attack (trial decryption

by an unauthorized user) will be increased.

For  $k = 1$ , equation (4) is same as the original Arnold transform of equation (1).

### 1.3. Fibonacci Transform

This is a special case of the basic Arnold transform, the equation of the transform is:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} [\text{mod } N] \quad (5)$$

Generally, the Fibonacci sequence  $F_n$ , is a sequence of integer numbers that given by equation (6):

$$F_n = \begin{cases} 0 & n=0 \\ 1 & n=1 \\ F_{n-1} + F_{n-2} & \text{otherwise} \end{cases} \quad (6)$$

By applying equation (6) the Fibonacci series constitutes the numbers:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233 ...

It can be easily seen that a 2x2 matrix formed by any four consecutive terms of the Fibonacci Numbers is a matrix and can be considered as an image scrambler. A generalized Fibonacci Transform can be defined as a mapping  $\Gamma: T^2 \rightarrow T^2$  such that:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} F_i & F_{i+1} \\ F_{i+2} & F_{i+3} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} [\text{mod } N] \quad (7)$$

Where,  $x, y, x', y' \in \{0, 1, 2 \dots N-1\}$ ,  $F^i$  is the  $i$ th term of the Fibonacci Series. And  $N$  is the zero size of a digital square image.

### 1.4. Lucas Series

Lucas series is a special case of Fibonacci series can be defined as follows:

$$L_n = \begin{cases} 2 & n=0 \\ 1 & n=1 \\ L_{n-1} + L_{n-2} & \text{otherwise} \end{cases} \quad (8)$$

By applying equation (8) the Lucas series constitutes the numbers:

2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199, 322, 521 ...

Unlike the Fibonacci series, the terms of the Lucas series does not form a periodic map and cannot be used for image encryption but by combining the terms of the  $N$  Fibonacci and Lucas series, a series of new periodic maps will be formed which can be used for image scrambling.

### 1.5. Fibonacci-Lucas Transform

The equation of this transform is:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} F_i & F_{i+1} \\ L_i & L_{i+1} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} [\text{mod } N] \quad (9)$$

Where,  $x, y, x', y' \in \{0, 1, 2 \dots N-1\}$ ,  $F^i$  is the  $i$ th term of the  $F12$  series,  $L^i$  is the  $i$ th term of the Lucas series, and  $N$  is the size of a digital squared image.

Continuing in this way we can form an infinitely many transforms and just like the Modified Arnold and Fibonacci Transforms, all of these matrices will be periodic in nature with a maximum possible periodicity of  $N^2-1$ .

## 2. The Proposed System

The proposed system consist of two stages, the first stage for encryption information and the second one for decryption, each one of these stages consist of many steps. The encryption steps will be illustrated in figures (3), which are:

- Translate image into  $YC_bC_r$  format and separating colored image into three grayscale image, each one of them represent one band like  $y, C_b, C_r$ .
- Applying normalization process for all pixels in each band according to the key value.
- Cipherring each pixel in first grayscale image ( $y$  band) by using receiver public key.
- Cipherring each pixel in second grayscale image ( $C_b$  band) by using receiver public key and sender private key.
- Cipherring each pixel in third grayscale image ( $C_r$  band) by using received public key.
- Scrambling all pixels in first band by using Arnold transform
- Scrambling all pixels in second band by using Fibonacci transform.
- Scrambling all pixels in third band by using Fibonacci-Lucas transform
- Reconstructing colored image.
- Scrambling all pixels in colored by using new Arnold transform.

While the decryption steps illustrated in figure (4), which are:

- Applying inverse Arnold transform on colored image.
- Separating cipherrd image into three grayscale image (bands).
- Applying inverse Arnold transform on first band.
- Applying inverse Fibonacci transform on second band.
- Applying inverse Fibonacci-Lucas transform on third band.
- Decipherring each pixel in first grayscale image by using receiver private key.
- Decipherring each pixel in second grayscale image by using receiver private key and sender public key.
- Decipherring each pixel in third grayscale image by using received private key.
- Return original values before normalization.
- Reconstructing  $YC_bC_r$  image.
- Translate to original .BMP image.

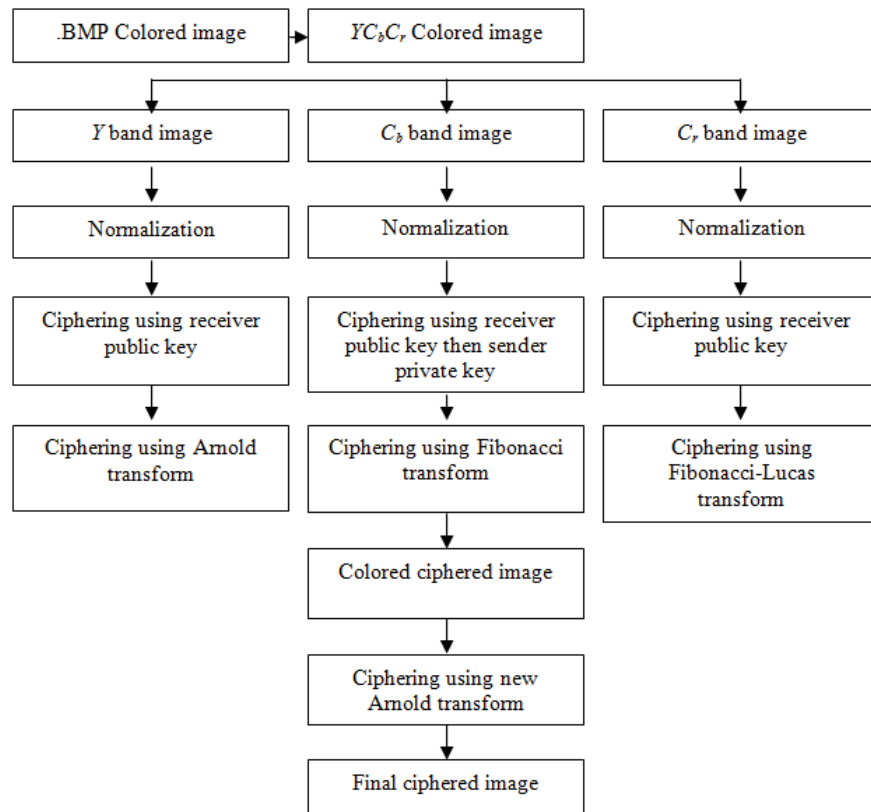


Figure (3). Encryption stage of the proposed system.

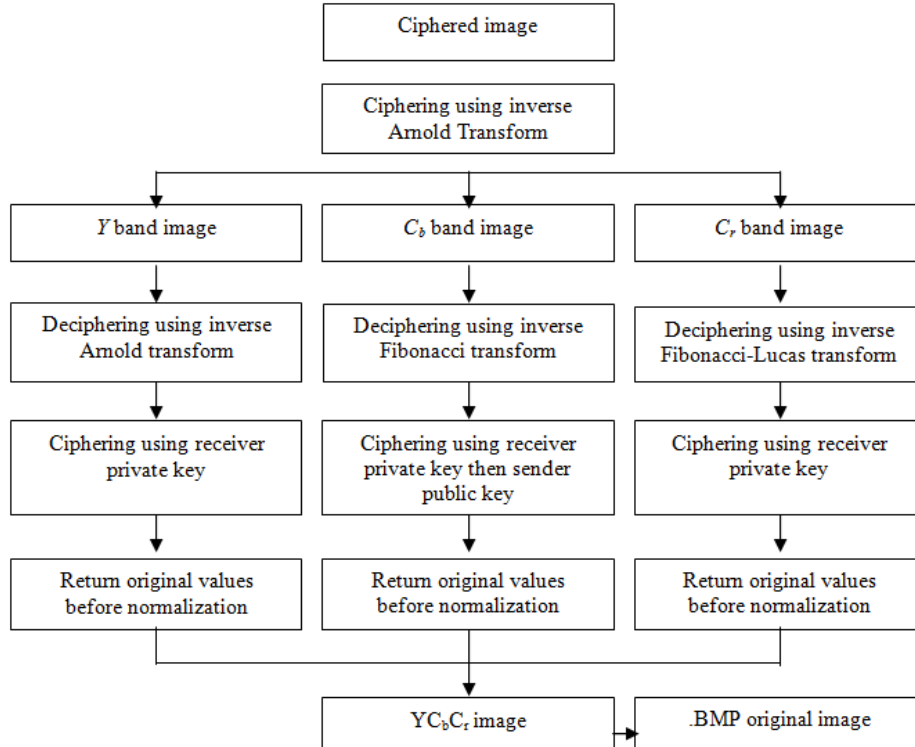


Figure (4). Decryption steps of the proposed system.

Each step in the proposed system will be illustrated with an example, by using the following keys:

- receiver public key (3,187),
- receiver private key (107,187),
- sender public key (11, 133), and
- sender private key (59, 133).

With the following bands for part of colored image as illustrated in figure (5).

3	200	210	150	80	100	250	150	30
30	100	250	22	66	120	100	40	20
40	150	255	8	78	250	90	20	10
a. red band			b. green band			c. blue band		

Figure (5). Bands for part of colored image.

### 3. First Stage

This stage for encryption information contains many steps as follow:

#### 3.1. Translating to $YC_bC_r$ Image



a- BMP image

b-  $YC_bC_r$  image

Figure (6). Translating to  $YC_bC_r$  image.

( $YC_bC_r$ ) image will be created from BMP image by using equation (10) as illustrated in figure (6) and (7):

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168 & -0.331 & 0.5 \\ 0.5 & -0.418 & -0.081 \end{bmatrix} \times \begin{bmatrix} red \\ green \\ blue \end{bmatrix} \quad (10)$$

To calculate values of all pixels in each band from red, green, and blue bands, Equation (10) will be applied as follows:

For y band:  $16+(0.299 \times red + 0.587 \times green + 0.114 \times blue)$ .  
For  $C_b$  band:  $128+(-0.168 \times red - 0.331 \times green + 0.5 \times blue)$ . For  $C_r$  band:  $128+(0.5 \times red - 0.418 \times green - 0.081 \times blue)$ .

The results of applying these states will be applied in figure (7).

133	140	141	203	143	75	47	182	189
49	89	163	173	109	56	126	147	201
43	109	240	164	87	7	137	169	150

Figure (7).  $YC_bC_r$  image values.

#### 3.2. Normalization

Normalization process is an important step to prepare image pixels for cipherring because we cannot apply modulo of (133) on 256 values without loose information, so all values must be less than (133) to insure information retrieve, these process applied on all bands according to value of (n) in the key. The y band will be encrypted by key (3, 187) so all pixels in y band must have values between 0 and 186, in the same way blue band will be encrypted with key (3, 187) so all pixels in  $C_r$  band must have values between 0 and 186, while the  $C_b$  band will be encrypted firstly by key (3, 187) so all pixels in y band must have values between 0 and 186, after that the ciphered pixels in  $C_b$  band will be ciphered by

key (59, 133) so all pixels in ciphered  $C_r$  band must have values between 0 and 132.

For y band:  $133 \times 186 / 255 = 97$ ,  $140 \times 186 / 255 = 102$ , and so on for other values. For  $C_r$  band:  $47 \times 132 / 255 = 24$ ,  $182 \times 132 / 255 = 94$ , and so on for other values.

And for  $C_b$  band: the first normalized process is  $203 \times 186 / 255 = 148$ ,  $143 \times 186 / 255 = 104$ .

The second normalized process will be done after cipherring with key (3, 187). The pixels value of colored image in figure (7) will be normalized as illustrated in figure (8).

The results of applying this step will be illustrated in figure (9).

97	102	103	148	104	55	24	94	98
36	65	110	126	80	41	65	76	104
31	80	175	120	63	5	71	87	78
a. Y band			b. $C_b$ band			c. $C_r$ band		

Figure (8). Bands of normalization Image.



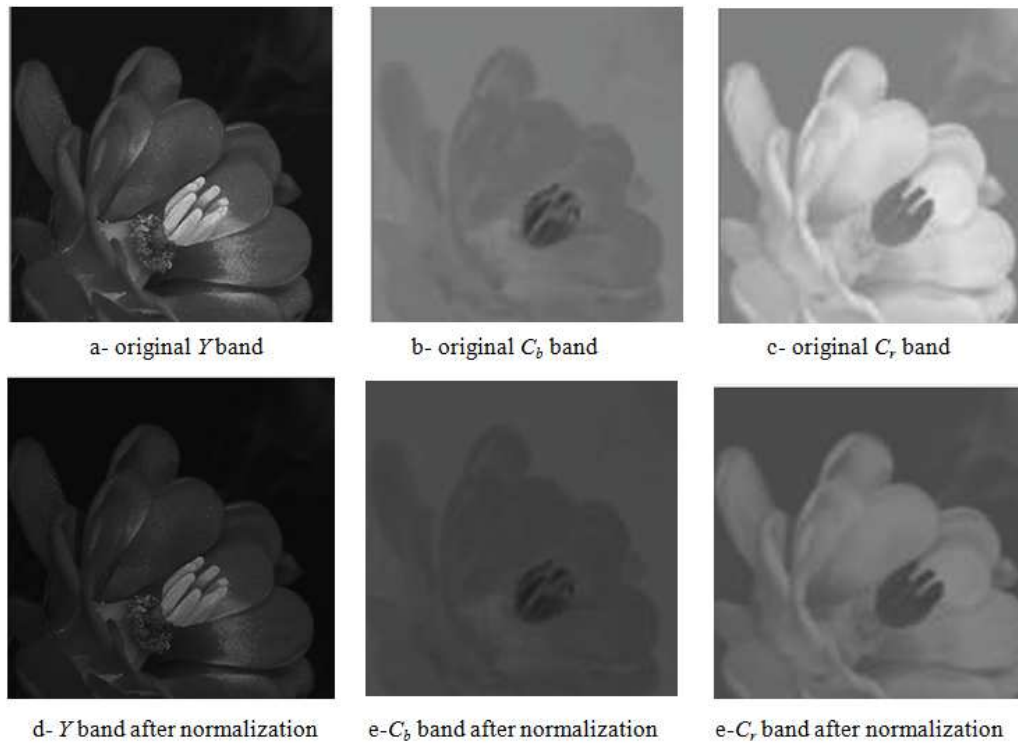


Figure (9). Results of normalization step.

### 3.3. Public Key Encryption

As illustrated in the proposed system block diagram, each band will be ciphered with different key or keys, so this step will be separated into three states.

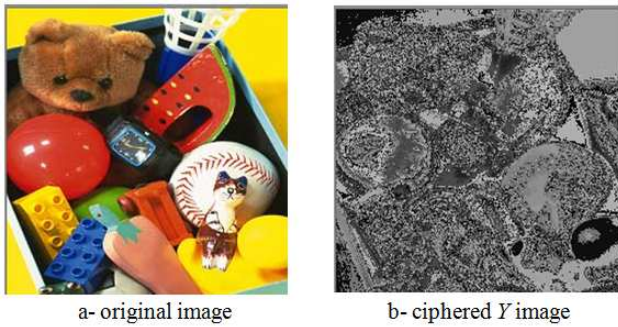


Figure (10). Encryption Y image.

$$C(97) = (97)^3 \bmod 187 = 113$$

97	102	103
36	65	119
31	80	175

a. original image

113	170	86
93	109	102
58	181	142

b. ciphered  $C_r$  image

Figure (11). Results after applying key (3, 187) on Y image.

#### 3.3.1. Y Band

In this band, the cipher used received public key for encryption all pixels in this band by applying equation (1) as

illustrated in figure (10) as well as in calculating the new ciphered values for supposed example with key (3, 187) as shown in figure (11).

#### 3.3.2. $C_b$ Band

In this band, the cipher used received public key and sender private key for encryption all pixels in this band by applying equation (1) with additional normalization process as illustrated in figure (12) as well as in calculating the new ciphered values for supposed example with keys (3, 187) and (59, 133) as shown in figure (13).

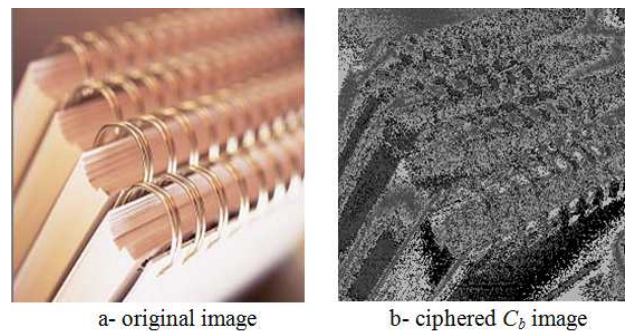


Figure (12). Encryption  $C_b$  image.

$C(97) = (148)^3 \bmod 187 = 147$  then normalized all values to cipher with key (59, 133)

$$147 \times \left( \frac{132}{255} \right) = 76$$

$$C(76) = (76)^{59} \bmod 133 = 76$$

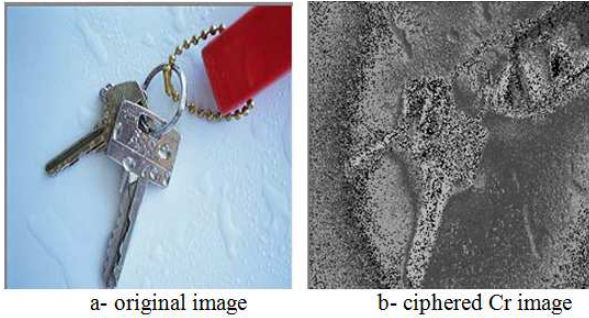
148	104	55	147	59	132	76	31	98	76	103	45
126	80	41	37	181	105	19	94	54	38	75	80
120	63	5	120	28	125	62	14	65	104	105	88

a. original image      b. first ciphered image      c. normalized image      d. final ciphered image

**Figure (13).** Results ciphering  $C_b$  image.

### 3.3.3. $C_r$ Band

In this band, the cipher used sender private key for encryption all pixels in this band by applying equation (1) as illustrated in figure (14) as well as in calculating the new ciphered values for supposed example with key (59, 133) as shown in figure (15).



**Figure (14).** Encryption  $C_r$  image.

97	102	103	113	170	86
36	65	119	93	109	102
31	80	175	58	181	142

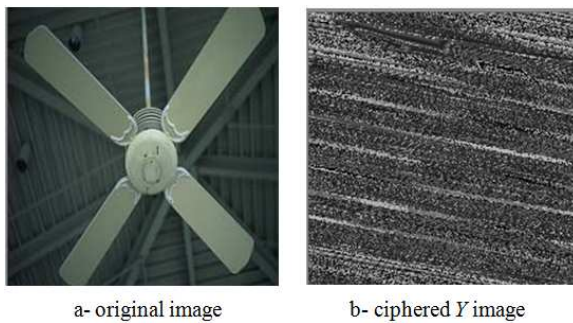
a. original image      b. ciphered  $C_r$  image

**Figure (15).** Results after applying key (3, 187) on  $C_r$  image.

### 3.3.4. Scrambling Pixels in Each Band

As illustrated in proposed system block diagram, each band will be scrambled with specific transform, so this step will be separated into three states.

#### 3.3.4.1. $Y$ Band



**Figure (16).** Scrambling  $Y$  image.

In this band, the cipher used Arnold transform for scrambling all pixels by applying equation (4) as illustrated in figure (16) as well as in calculating the new ciphered values for supposed example as shown in figure (17).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 6 & 5 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \pmod{3} = (0, 0),$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 6 & 5 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \pmod{3} = (2, 1)$$

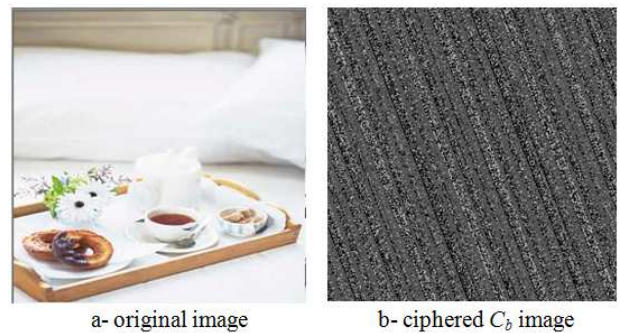
97	102	103	97	36	31
36	65	119	119	175	103
31	80	175	80	170	65

a. original image      b. ciphered  $Y$  image

**Figure (17).** Results after scrambling  $Y$  image.

#### 3.3.4.2. $C_b$ Band

In this band, the cipher used Fibonacci transform for scrambling all pixels in this band by applying equation (7) as illustrated in figure (18) as well as in calculating the new ciphered values for supposed example as shown in figure (19).



**Figure (18).** Scrambling  $C_b$  image.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 8 & 13 \\ 21 & 34 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \pmod{3} = (0, 0),$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 8 & 13 \\ 21 & 34 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \pmod{3} = (1, 1)$$

76	103	45
38	75	80
104	105	88

a. original image

76	75	88
104	103	80
38	105	45

b. ciphered  $C_b$  image**Figure (19).** Results after scrambling  $C_b$  image.

113	170	86
93	109	102
58	181	142

a. original image

113	102	181
142	170	93
109	58	86

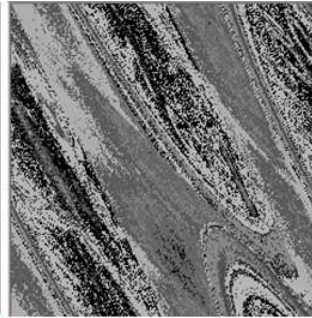
b. ciphered  $C_r$  image**Figure (21).** Results after scrambling  $C_r$  image

### 3.3.4.2. $C_r$ Band

In this band, the cipher used Fibonacci-Lucas transform for scrambling all pixels in this band by applying equation (9) as illustrated in figure (20) as well as in calculating the new ciphered values for supposed example as shown in figure (21).



a- original image

b- ciphered  $C_r$  image**Figure (20).** Scrambling  $C_r$  image.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \pmod{3} = (0,0),$$

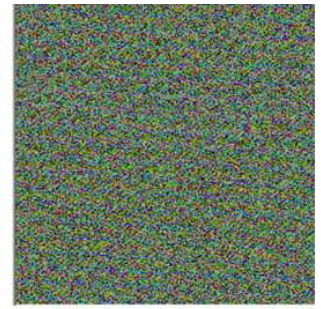
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \pmod{3} = (1,1)$$

### 3.3.5. Scrambling All Pixels in Colored Image

In this step, the cipher used Arnold transform for scrambling all pixels in this band by applying equation (4) as illustrated in figure (22) as well as in calculating the new ciphered values for supposed example as shown in figure (23).



a- original image



b- ciphered colored image

**Figure (22).** Scrambling Colored Image.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 20 & 19 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \pmod{3} = (0,0),$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 20 & 19 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \pmod{3} = (1,1)$$

97	36	31
119	175	103
80	170	65

a.  $Y$  image

76	75	88
104	103	80
38	105	45

b.  $C_b$  image

113	102	181
142	170	93
109	58	86

c.  $C_r$  image

97	65	175
103	36	80
170	119	31

d. final scrambling  $Y$  image

76	45	103
80	75	38
105	104	88

e. final scrambling  $C_b$  image

113	86	170
93	102	109
58	142	181

f. final scrambling  $C_r$  image**Figure (23).** Scrambling colored image.

### 3.4. Second Stage

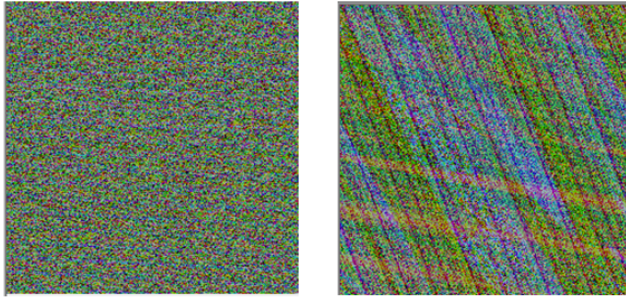
The input for this stage is the final ciphered image that

result from first stage and this image will be passed through the same processes in the first stage but in reverse ordering as follows:



### 3.4.1. Applying Inverse Arnold Transform

In this step, the inverse Arnold transform will be applied in colored image as illustrated in figure (24).



a- ciphered image

b- ciphered image after applying Arnold transform

Figure (24). Applying inverse Arnold transform on colored image.

97	65	175
103	36	80
170	119	31

76	45	103
80	75	38
105	104	88

113	86	170
93	102	109
58	142	181

a. bands of ciphered image.

97	36	31
110	175	103
80	170	65

76	75	88
104	103	80
38	105	45

113	102	181
142	170	93
109	58	86

b. bands after applying inverse Arnold transform.

Figure (25). Applying inverse Arnold transform on colored image.

### 3.4.2. Return Pixels in Each Band to Its State before Scrambling

As illustrated in proposed system block diagram, each band will be decryption with specific inverse transform, so this step will be separated into three states.

#### 3.4.2.1. Y Band

In this band, the decipher used inverse Arnold transform for return all pixels in this band to its past original state before scrambling as illustrated in figure (26).

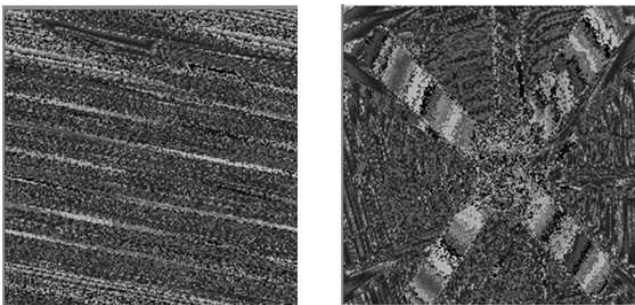


Figure (26). Applying inverse Arnold transform on Y image.

The inverse Arnold transform can be illustrated in figure (25) by using equation below:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & -19 \\ -1 & 20 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} [\text{mod } 3] = (0, 0),$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & -19 \\ -1 & 20 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} [\text{mod } 3] = (2, 2)$$

Calculating values before scrambling in Y band for supposed example will be shown in follows and in figure (27).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & -5 \\ -1 & 6 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} [\text{mod } 3] = (0, 0),$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & -5 \\ -1 & 6 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} [\text{mod } 3] = (1, 0)$$

97	36	31
119	175	103
80	170	65

97	102	103
36	65	119
31	80	175

Figure (27). Applying inverse Arnold transform on Y image.

#### 3.4.2.2. C<sub>b</sub> Band

In this band, the decipher used inverse Fibonacci transform for return all pixels in this band to its past original state before scrambling as illustrated in figure (28).

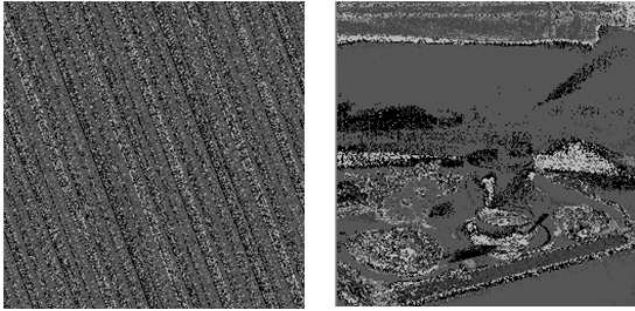


Figure (28). Applying inverse Fibonacci transform on  $C_b$  image.

Calculating values before scrambling in  $C_b$  band for supposed example will be shown in follows and in figure (29).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -34 & 13 \\ 21 & -8 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} [\text{mod } 3] = (0, 0),$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -34 & 13 \\ 21 & -8 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} [\text{mod } 3] = (1, 0)$$

76	75	88
104	103	80
38	105	45

76	103	45
38	75	80
104	105	88

Figure (29). Applying inverse Fibonacci transform on  $C_b$  image.

### 3.4.2.3. $C_r$ Band

In this band, the decipher used inverse Fibonacci-Lucas transform for return all pixels in this band to its past original state before scrambling as illustrated in figure (30).

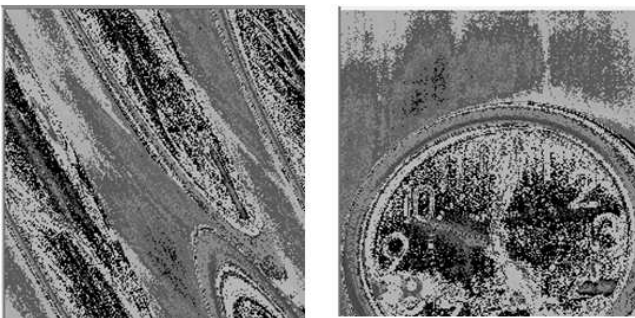


Figure (30). Applying inverse Fibonacci-Lucas transform on  $C_r$  image.

Calculating values before scrambling in  $C_r$  band for supposed example will be shown in follows and in figure (31).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} [\text{mod } 3] = (0, 0),$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} [\text{mod } 3] = (1, 2)$$

113	102	181
142	170	93
109	58	86

113	170	86
93	109	102
58	181	142

Figure (31). Applying inverse Fibonacci-Lucas transform on  $C_r$  image.

### 3.4.3. Public Key Decryption

As illustrated in proposed system block diagram, each band will be deciphered with different key or keys, so this step will be separated into three states.

#### 3.4.3.1 Y Band

In this band, the decipher used received private key for decryption all pixels in this band by applying equation (2) as illustrated in figure (32) as well as in calculating the new ciphered values for supposed example with key (107, 187) as shown in figure (33).

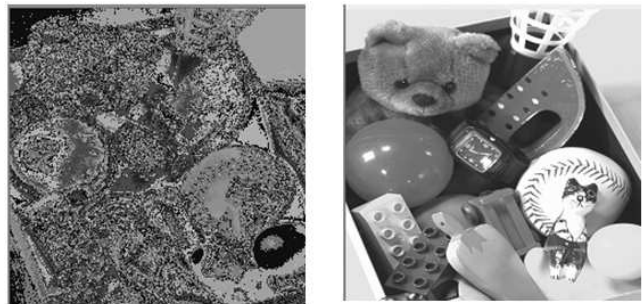


Figure (32). Decryption Y image.

$$P_{(113)} = (113)^{107} \text{ mod } 187 = 113$$

113	170	86
93	109	102
58	181	142

97	102	103
36	65	119
31	80	175

Figure (33). Results after applying key (107, 187) on Y image.

#### 3.4.3.2. $C_b$ Band

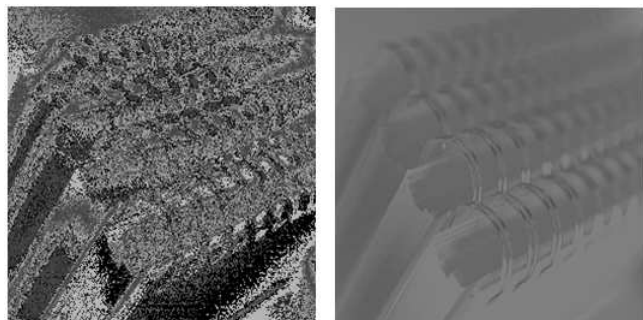


Figure (34). Decryption  $C_b$  image.

In this band, the decipher used sender public key and received private key for decryption all pixels in this band

by applying equation (2) with additional inverse normalization process as illustrated in figure (34) as well as in calculating the new deciphered values for supposed example with keys (107, 187) and (11, 133) as shown in figure (35).

$P(76)=(76)^{11} \bmod 133=76$  then normalized all values to cipher with key (11, 133)

76	103	45	76	31	68	147	59	132	148	104	55
38	75	80	19	94	54	37	181	105	126	80	41
104	105	88	62	14	65	120	28	125	120	63	5
a. ciphered image			b. first deciphered image			c. normalized image			c. final deciphered image		

Figure (35). Results deciphering  $C_b$  image.

#### 3.4.3.3. $C_r$ Band

In this band, the decipher used sender public key for decryption all pixels in this band by applying equation (2) as illustrated in figure (36) as well as in calculating the new ciphered values for supposed example with key (107, 187) as shown in figure (37).

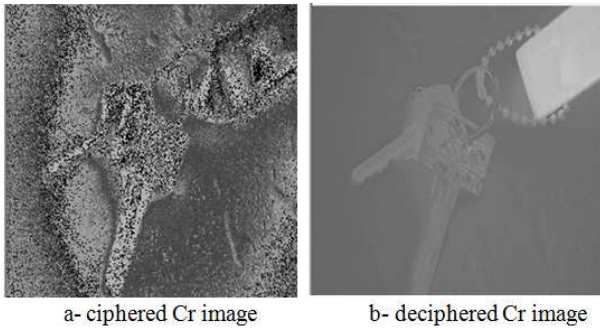


Figure (36). Decryption  $C_r$  image.

$$P(113)=(113)^{107} \bmod 187=97$$

113	170	86	97	102	103
93	109	102	36	65	119
58	181	142	31	80	175

Figure (37). Results after applying key (107, 187) on  $C_r$  image.

#### 3.4.4. Return Original Values before Normalization

In this step inverse normalized ratio will be applied to each pixel in each band as follows:

For y band: pixel value $\times 255/186$ , for  $C_r$  band: pixel value $\times 255/132$ , and for

$C_b$  band: pixel value $\times 255/186$ . this process results will be illustrated in figure (38).

97	102	103	148	104	55	24	94	98
36	65	119	126	80	41	65	76	104
31	80	175	120	63	5	71	87	78
a. bands after normalization.								
133	140	141	203	134	75	47	182	189
49	89	163	173	109	56	126	147	201
43	109	240	164	87	7	137	169	150
b. bands before normalization.								

Figure (38). Applying inverse normalization process.



### 3.4.5. Translating to BMP Image

To translate an image from YCbCr format to BMP format, inverse array used in equation (10) will be used as follows in equation (11) the result of applying this equation will be illustrated in figure (39).

$$\begin{bmatrix} \text{red} \\ \text{green} \\ \text{blue} \end{bmatrix} = \begin{bmatrix} 0.9986 & -0.0004 & 1.4027 \\ 1.0011 & -0.3439 & -0.7142 \\ 0.9982 & 1.7722 & -0.0015 \end{bmatrix} \begin{bmatrix} Y-16 \\ C_b-128 \\ C_r-128 \end{bmatrix} \quad (11)$$

133	140	141
49	89	163
43	109	240

a.  $Y$  band

203	143	75
173	109	56
164	87	7

b.  $C_b$  band

47	182	189
126	147	201
137	169	150

c.  $C_r$  band

3	200	210
30	100	250
40	150	255

d. red band

150	80	100
22	66	120
8	78	250

e. green band

250	150	30
100	40	20
90	20	10

f. blue band

Figure (39). Translating  $YC_bC_r$  image into BMP image.

## 4. Results

After applying the proposed system on colored image, the following results will be founded.

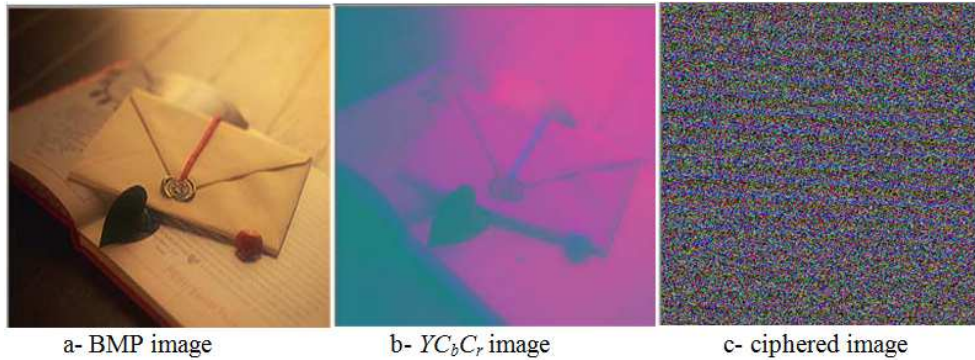


Figure (40). Result (1) of the proposed system.

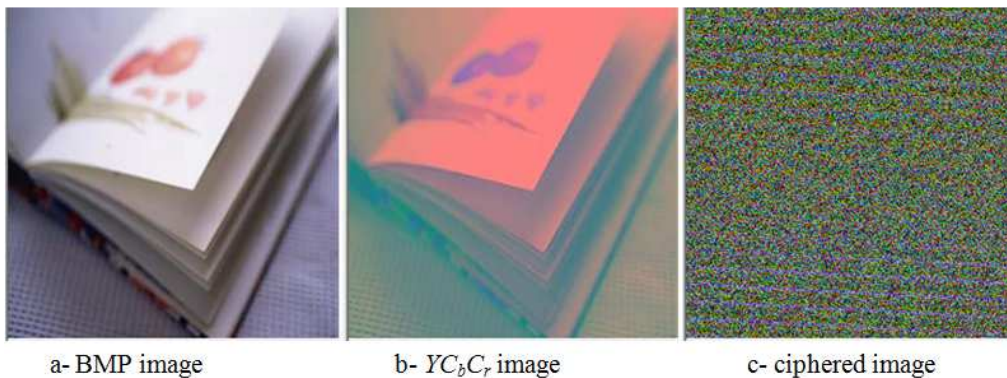


Figure (41). Result (2) of the proposed system.



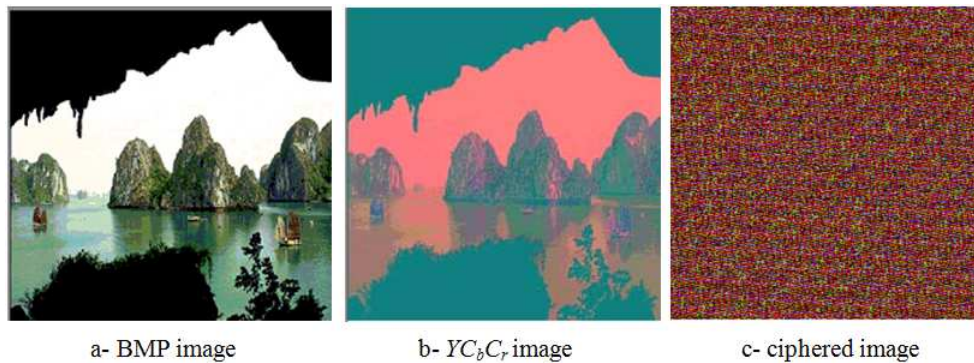


Figure (42). Result (3) of the proposed system

## 4. Conclusion

From the above discussions and results we conclude that:

- Scrambling image pixel is a powerful tool in ciphering image and it make image unreadable and unproductive, but scrambling methods must have long randomness cycle to avoid repeated values of new position for all pixels in original image.
- Public key encryption changes the pixels value in image in fashion that make the image can be predictive with huge amount of noise, so this method cannot be used alone.
- Normalization process is an important step to make information space identical to key space without losing information that damage image.

- [2] Kwok-Wo Wong, Bernie Sin-Hung Kwok, and Wing-Shing Law " A Fast Image Encryption Scheme based on Chaotic Standard Map", HONG KONG, 2006.
- [3] Abdullah Sharaf Alghamdi, and Hanif Ullah "A Secure Iris Image Encryption Technique Using Bio-Chaotic Algorithm", IJCNS, 2010.
- [4] Minati Mishra, Priyadarsini Mishra, M.C. Adhikary and Sunit Kumar," Image Encryption Using Fibonacci-Lucas Transformation", IJCIS, India, 2012.
- [5] Rajinder Kaur, Er.Kanwalprit Singh " Image Encryption Techniques: A Selected Review", IOSR-JCE, India, 2013.
- [6] Quist-Aphetsi Kester " Image Encryption Based on the RGB PIXEL Transposition and Shuffling" I. J. Computer Network and Information Security, 2013.

## References

- [1] Linhua Zhang , Xiaofeng Liao , Xuebing Wang " An Image Encryption Approach Based on Chaotic Maps", .elsevier , China, 2004.

## Biography



**Dr. Salman Abd Kadum, Ph. D., Ms.C.** Technical Education Foundation, Najaf Technical Institute, Najaf, IRAQ. He received B.Sc. in Physics sciences (1983) from Baghdad university, High Deploma in computer science (1988) from training & research institute, Baghdad, Iraq, M.Sc. in computer sciences (1998) from Gajah Mada

university, Indonesia, and Ph. D. in computer science its also from Gajah Mada University, Indonesia (2004), worked as a lecturer as long time and a head of computer system in Najaf Technical Institute for two years, he is also a master of computer center for two years, and he is also as lecturer for eight years and a head of software engineering in Al-Imam Jaafar Alsadeq university for five years, he has many published papers. His researches interests are in image processing, and signal processing.



**ALI ADUL AZEEZ MOHAMMAD BAKER** University of Kufa, Education College, Najaf, IRAQ. He received B.Sc. in computer sciences (2010), civil engineering (1990), and M.Sc. in computer sciences (2012), worked as a teacher in Kufa University, has many published papers. His research interests

are in image processing, and security (biometrics, and steganography), He's Associate teacher in Computer Science at the University of Kufa – Najaf, IRAQ.