A MODIFIED METHOD OF INFORMATION HIDING BASED ON HYBRID CRYPTOGRAPHY AND STEGANOGRAPHY

Fadhil Salman Abed
Iraq, Diyala, Polytechnic University-Sulaimni -
Technical Institute of Kalar
,E-Mail Fad_Sal_Abed@Yahoo.com

Abstract

"Combination of cryptography and Steganography for secure communication" is an application, which combines both Cryptography methods (i.e. Encryption, decryption) and Steganography techniques to make the communication more secure. The outcome of this project is to create a cross-platform tool that can effectively hide a message (i.e. Word document) inside a image file. It is concerned with embedding information in a secure and robust manner.

The proposed system depends upon preparing the image data for the next step (DCT Quantization) through steganographic process and using two levels of security: the hybrid RSA with Knapsack algorithm, A hybrid of RSA and knapsack cryptosystem can be represented as a secure system . The results that achieved from using this technique is the encrypted message which acquires a higher security and accuracy when compared with RSA and knapsack cryptosystem individually in modern communication systems to protect the information transmitted over an insecure communication channel .

then storing the image in a JPEG format. In this case, the secret message will be looked as plaintext with digital signature while the cover is a coloured image. Then, the results of the algorithm are submitted to many criteria in order to be evaluated that prove the sufficiency of the algorithm and its activity. Thus, the proposed algorithm for this research can be divided into two main parts: hiding the text of the sender, and extracting it by the receiver.

Keywords

DCT Transformation, Public key encryptions, Image processing , Hiding, Cryptography, Steganography

1.0 Introduction

With the development of internet technologies, digital media can be transmitted conveniently over the internet. However, messages transmission over the internet still have to face all kinds of security problems. Neither Cryptography nor Steganography is a turnkey solution to privacy of open systems. To add multiple layers of security it is always a good practice to use both Cryptography and Steganography together[1].

Therefore, how to protect secret messages during transmission becomes an essential issue for the internet. The schemes include DES, AES and RSA**[2].** These methods scramble the secret message so that it cannot be understood. However, it makes the messages suspicious enough to attract eavesdroppers attention hence, this paper presents a novel two techniques

are available to those wishing to transmit secrets using unprotected communications media. One is cryptography, where the secret is scrambled and can be reconstituted only by the holder of a key. The second method is steganography, where the secret is encoded in another message in a manner such that, to the casual observer, it is unseen. Steganography is often combined with cryptography to provide an additional layer of security.

The JPEG format is currently the most common format for storing image data. It is also supported by virtually all software applications that allow viewing and working with digital images. Recently, several steganographic techniques for data hiding in JPEG have been developed**[3].**

The proposed algorithm depends on preparing the image data for next steps (DCT, quantization) through embedding processes and using two levels of security hybrid RSA with Knapsack cryptosystem, later the stego image is JPEG format. The secret message in this approach is plaintext, digital signature, while the cover is a coloured image. The algorithm results are submitted to many evaluating criteria to evaluate them, which prove their efficiency and activity. The proposed algorithm of this paper can be divided into two main parts, hiding and extracting, each of them can be further divided into a number of procedures**[4].**

2. The Proposed Hiding System

This stage includes two parts; cryptography and steganography. In our research we Proposed a hybrid combination of two public-key cryptosystem RSA and Knapsack, which offers extremely good security with less complexity and less time required for encryption- decryption process in compression with RSA and knapsack individually. The idea behind of this system is to use two stages of encryption. The first stage is a knapsack cryptosystem and the second stage is the RSA cryptosystem as shown in the figure(1). The enciphered text at the output of the first stage is as input message for the second stage. In the receiver, the enciphered transmitted message is decrypted using the RSA algorithm firstly and the knapsack algorithm secondly. Thus the decryption order is opposite to the encryption order. The steganography part includes hiding the secret message and digital signature after converting them to stream of bits.

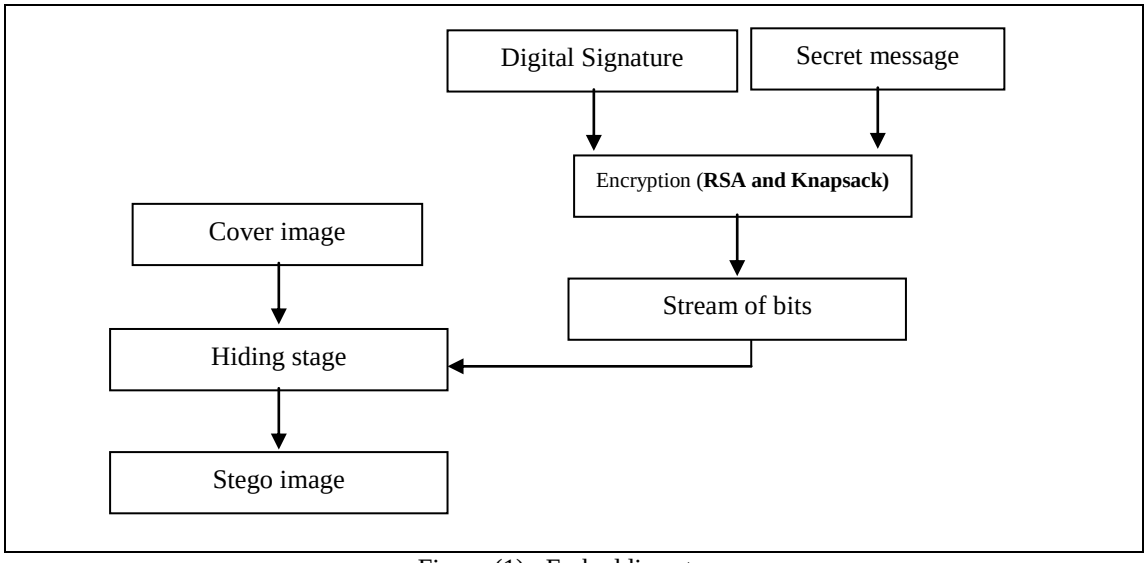


Figure (1): Embedding stage

2.1 Cryptography Stage

There are several types of asymmetric algorithms used in the computing world today. They may have different internal mechanisms and methods, but the one thing they do have in common is that they are all asymmetric. This means that a key is used to encrypt a message different from the key that is used to decrypt a message. RSA is a public key algorithm that is the most understood, easiest to implement, and the most popular when it comes to asymmetric algorithm.

2.1.1 The Proposed System

It is well Known that the most important topic in the filed of communication systems including internet, satellite or mobile is a security of exchanging the sensitive information. The previous two ciphering techniques are maintained, RSA and knapsack cryptosystems. The security of RSA cryptosystem is based on the problem of factoring a large number. A factoring of n would enable the eavesdropper to break the algorithm. The factors of n enables the eavesdropper to compute $\phi(n)$, and thus d. The security of the knapsack cryptosystem is based on finding a binary vector (x_1, x_2, …., x_n) such that

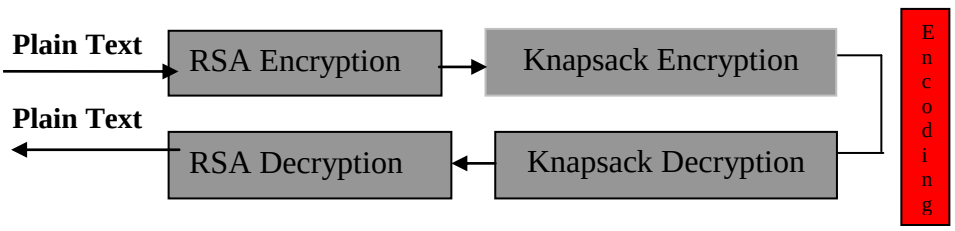
$$\mathbf{Y}=\sum_{j=1}^{n} a_j x_j$$

Where Y is is the information transmitted over an insecure channel so that any eavesdropper has access to it. The only solution available is an enumeration method search over all 2^n possible vector of X.

For a highly secure cryptosystem, it is necessary to use a long n in the RSA cryptosystem, or a long knapsack vector (x_1, x_2, …., x_n) in the knapsack cryptosystem. However, increasing the security yields the following disadvantages:

1. Complexity of the system increases,
2. Requirement on the memory size increases,
3. Encryption an decryption time increases,
4. Add requirement of multi-precision arithmetic and fast algorithms,
5. Requirement of searching for a long prime number.

To overcome this problem, a hybrid combination is suggested as shown in Figure(2), which offers extremely good security with less complexity and less time required for encryption- decryption process in compression with RSA and knapsack individually. The idea behind of this system is to use two stages of encryption. The first stage is a knapsack cryptosystem and the second stage is the RSA cryptosystem as shown in the figure. The enciphered text at the output of the first stage is as input message for the second stage. In the receiver, the enciphered transmitted message is decrypted using the RSA algorithm firstly and the knapsack algorithm secondly. Thus the decryption order is opposite to the encryption order. Figure(3) illustrated the main steps of the Proposed System.



Figure(2): The main steps of the Proposed System

RSA Cipher stage

- Chose prime number p nd q
- n= pxq
- ϕ (n)=(p-1) x (q-1)
- chose, where 1< e > ϕ
- compute d by using $d=e^{-1} \bmod \phi$, $c = c_{k(i)}^{e} \bmod n$

Knapsack ciphering stage

- Input knapsack sequence
- Chose superincreasing sequence which satisfy $a_{i+1}^{\bullet} \rangle \sum_{i=1}^{n} a_i^{\bullet}$
- Chose M , W , that $M \rangle \sum_{i=1}^{n} a_i^{\bullet}$, W< M , gcd(M, W)=1.
- Compute the public-key, by using the equation:- $a = a_i^{\bullet} \ . \ (\bmod M)$
- Convert the message X as a binary string of length n: X=x_1,x_2,x_3,……., xn
- The ciphering using the equation:-
- $C = \sum_{i=1}^{n} a_i . \ x_i$

Knapsack deciphering stage

- Calculate $C^{\bullet} = W^{-1} \ . \ C$
- **Algorithm** - to solve knapsack problems with super increasing vectors:

for $i \rightarrow n$ **downto** *2* **do** **if** $C^{\bullet} \rangle 2a_i^{\bullet}$ **then** terminate {no solution}

else if $C^{\bullet} \rangle a_i^{\bullet}$ **then** $x_i \leftarrow 1$; $C^{\bullet} = C^{\bullet} - a_i^{\bullet}$;

else $x_i = 0$;

if $C^{\bullet} = a_1^{\bullet}$ **then** $x_1 = 1$

else if $C^{\bullet} = 0$ **then** $x_1 \leftarrow 0$;

else terminate {no solution} $C^{\bullet}$

RSA deciphering stage

- $C_k = C_k^{d} \bmod n$

Figure(3): illustrated the main steps of the Proposed System

2.1.2 Example: (Illustration of the Proposed System by small numbers)

Suppose we chooses p = 5, and q = 7, e = 5
Compute n = pq =35, and ϕ = (p-1)(q-1) = (4)(6) = 24
Compute d as $d=e^{-1} \bmod 24=29$
public key is given by (n, e) = (35, 5);
the private key is given by (n, d) = (35, 29)

Suppose that each letter as a number between 1 and 26.
That is, a = 1, b = 2, c = 3,…, n = 14,…, z = 26.

Assume plain text is: Love

Plaintext letter	m (numeric representation)	m^e	$c = m^e \bmod n$
l	12	248832	17
o	15	759375	15
v	22	5153632	22
e	5	3125	10

Cipher text produce from RSA encryption is: 17152210

Knapsack ciphering stage

Chose
A'=(1,2,4,9,18,35,75,151,302,606) ,
M = 1250, W = 41

Compute public- key
A=(1,82,164,369,738,185,575,1191,1132,1096)
In order to encrypt an English plaintext, we first encode its letters by 5-bit numbers _ - 00000, A - 00001, B - 00010,… and then divide the resulting binary strings into blocks of length 10.
Cipher text produce from RSA encryption is: 17152210 = QOVJ
Which is equal to = **10001 01111 10111 01010** in binary representation.
The encoding by using Knapsack
c1=41*1+82*0+164*0+369*0+738*1+185*0+575*1+1191*1+1132*1+1096*1=4773
c1=41*1+82*0+164*1+369*1+738*1+185*0+575*1+1191*0+1132*1+1096*0=3019

Decryption the cipher text produce from Knapsack (4773, 3019)

By multiplying with W^{-1} = 61 (mod 1250) we get new crypto-texts (several new c')

C'_1=61*4773 (mod 1250)=291153 (mod 1250)=1153
By applying the decryption knapsack algorithm, we obtain **10001 01111**
C'_2=61*3019 (mod 1250)=184159 (mod 1250)=409
By applying the decoding algorithm , we optane **10111 01010**
Which is equal to = **10001 01111 10111 01010=** QOVJ= 17152210
By applying RSA decryption with d = 29, and n = 35 is as follows:

Cipher text produce from Knapsack encryption is: (4746, 3019)

Ciphertext	c^d	$m = c^d \bmod n$	Plaintext letter
17	481968572106750915091411825223071697	12	l
15	127834039488589391112327575683593 75	15	o
22	851643319086537701956194499721106030592	22	v
10	10000000000000000000000000000	5	e

Then the plain text produce form combination RSA with Knapsack cryptosystem is
Cipher text produce from Knapsack encryption is 1215225=LOVE

*2.2 Hiding stage(*Steganographic)

Each steganographic method has its own advantages and disadvantages. The 2-LSB method is clear, very simple to implement, and if the encoded image is transmitted perfectly with no error, when it is decoded, there will be no data lost in the text (digital signature and message). The disadvantages of the 2-LSB method lies in that if the form of stego image is changed in any way (resized or compression to JPEG), the entire text could be lost. The size of stego image is very high, therefore it needs more time when it is transmitted via the internet.

The above disadvantages lead to suggest a new method to embed text

in the cover image. The new method used covers image bitmap (24 bit) and produces stego image JPEG format. The stego image in our case is a small size, it needs less time when it is transmitted via the internet, and if it is changed (JPEG to bitmap or recompressed more than one time) the entire text will be kept.

2.2.1 *DCT (Discrete Cosine Transformation) Technique*

A more complex way of hiding a secret message inside an image comes with the use and modifications of DCT. DCT are used by the JPEG compression algorithm to transform successive 8 x 8 pixel blocks of the image, into 64 DCT coefficients.

In this work one coefficient is used from each block (8 x 8) to hold the bit called DC coefficient in position (0,0);by preparing the value of pixels in the block until the DC coefficient becomes odd or even dependent on the bit which is wanted to be hidden. **Figure (4)** shows the general description of embedding messages and digital signatures in an image. This work includes the following steps :

- Load a colour image (bitmap format 24 bits), and part the colours into the red, green, and blue.
- Convert the image formula from RGB to YCbCr **[5]**, **[6]**

$$Y = (77/256)\,R + (150/256)\,G + (29/256)\,B$$
$$Cb = -(44/256)\,R - (87/256)\,G + (131/256)\,B + 128$$
$$Cr = (131/256)\,R - (110/256)\,G - (21/256)\,B + 128$$

- Separate the image components Y, Cb, Cr into blocks, each one consists of 64 pixels (8x8)
- Transform each block (8x8) pixels to spatial frequency domain via the forward DCT

$$G_{ij} = \frac{1}{4} c_i c_j \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} P_{xy} \cos\left(\frac{(2x+1)i\pi}{16}\right) \cos\left(\frac{(2x+1)j\pi}{16}\right)$$

(P_{xy}) pixel in the block coefficients , (G_{ij}) DCT coefficient**[7]**.

- Combine the stream of bits of the digital signature and the secret message.
- Embed the stream of bits in the cover image. In each block embed, one bit in the DC element. This step will be described in details later on.

Quantize these blocks with quantization coefficients. The DCT coefficients are divided by their corresponding quantization coefficients (quantization table) and rounded to the nearest integer.

- Quantize DCT coefficients by multiplying the same quantization tables that are used in a compression stage to obtain DCT coefficients.
- Inversing DCT is applied in this step in each block**[7], [4]**

$$p_{xy} = \frac{1}{4} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} c_i c_j G_{ij} \cos\left(\frac{(2x+1)i\pi}{16}\right) \cos\left(\frac{(2y+1)j\pi}{16}\right),$$

where $$c_i c_j = \begin{cases} \frac{1}{\sqrt{2}} & \text{fori, j} = 0 \\ 1 & \text{Oterwise} \end{cases}$$

- 10.Reconstruct the image by combining all the blocks.
- 11.Transform the image formula from YCbCr to RGB **[5] [6]**

$$R = Y + 1.371\,(Cr - 128),$$
$$G = Y - 0.698\,(Cr - 128) - 0.336\,(Cb - 128)$$
$$B = Y + 1.732\,(Cb - 128)$$

- 12. Save the stego image in JPEG format.

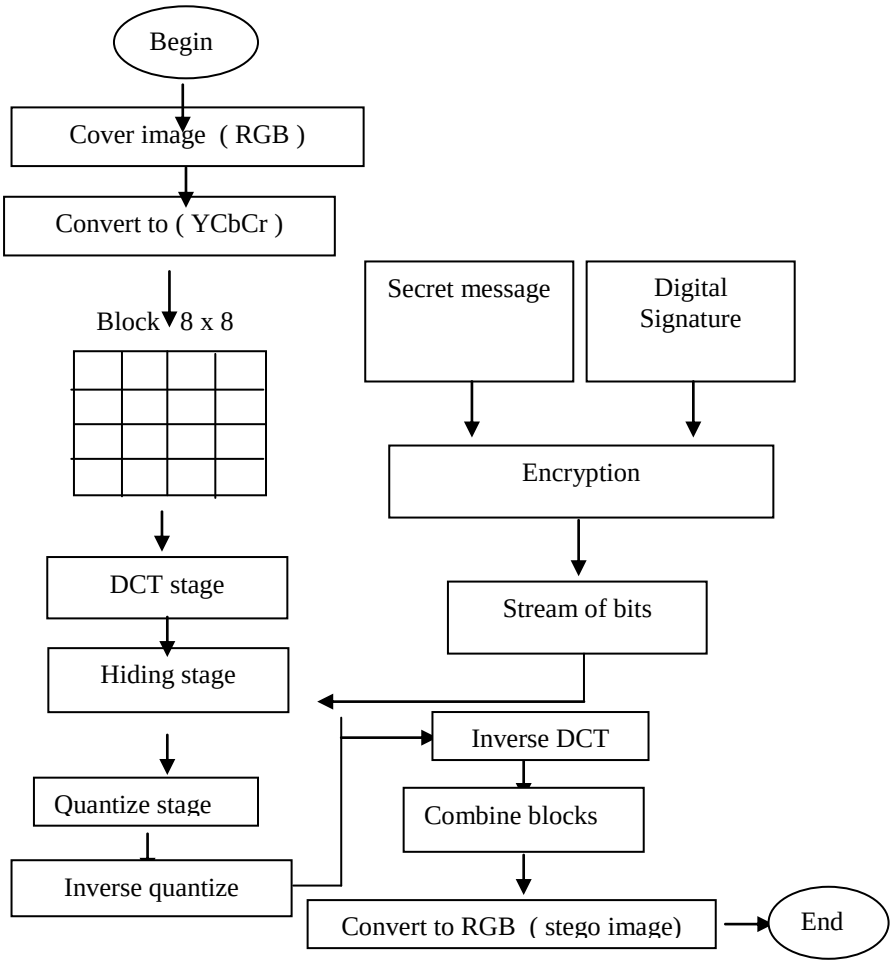


Figure (4): Encoding and hiding digital signatures and texts

2.2.2 *Hiding Bits*

In this step, the text bits are embedded in the cover-image. After inputting the text in the system, convert each letter in the text to a decimal number and encrypt it and convert each one to a binary form. , Some steps are implemented to embed the text bits. These steps are **[8]:**

1.From each block (8x8) one DCT coefficient is chosen to hold the bit. This coefficient is in position (0,0). Compute the quantize DC coefficient (**dc**) .

$$dc = \text{round}(\text{block}(0,0)/16)$$

2.If (**dc**) is an odd number and the bit ('1'), or (**dc**) is an even number and bit ('0'), no change happens in the original block pixels. Bring a new block and a new bit to continue or work and hide them.

3 .If (**dc**) is an odd number and the bit ('0'), or (**dc**) is an even number and bit ('1'), there must be a change in the original block coefficients until (**dc**) value satisfies the relationship in point (2). How can be satisfied in this work.

❖ Compute the new Quantize DC coefficient without rounding (**dc1**).

$$dc_1 = (block(0,0)/16$$

If (**dc > dc1**) then find the different (**df**) between them. To determine the number of pixels (**np**) that must be changed by **subtracting one** to

the original value through comparing (**np**) with table (1). Table (1) shows the number of pixels that must be changed, subtracted or added depending on the number of difference.

Df=dc-dcl

Np=0.5-df

If (dc1 > dc) then find the different (***df***) between them. To determine the number of pixels (***np***) that must be changed by ***adding one*** to the original value through comparing (***np***) with a table which contains how many numbers that must be subtracted or added corresponding to the difference between the two results.

df=dc1-dc ,
np=0.5-df

Table (1): Shows the amount added or subtracted of each block

Difference	Amount
0.000-0.063	**8**
0.064-0.125	**16**
0.126-0.188	**24**
0.189-0.250	**32**
0.251-0.313	**40**
0.314-0.375	**48**
0.376-0.437	**56**
0.438-0.500	**64**

- ❖ After subtracting or adding from the original coefficients block work, the research employs the DCT in the same block again.

4. Steps (1-3) continue with each block until hiding all the bits.

2.2.3 *Information about the Extracting Stage*

This stage includes two parts, first extract bits from a stego image and convert each group of bits (12 bits) to a decimal number, second decrypt the decimal number to find the digital signature and the message. Figure (5) shows the general description of extracting messages and digital signatures from images**[9]**. The extracting message and digital signature stage include the following steps :

1. Load the stego image (bitmap 24 bits). This image contains the digital signature and the secret message.
2. Convert the image formula from RGB to YC_bC_r.
3. Separate image components into blocks, each one consists of (8 x 8) pixels.
4. Transform each block (8 x 8) pixels to spatial frequency domain via the forward DCT. This step is executed on the Y components only.
5. The fifth step includes:
   - ♦ Extract the digital signature bits and convert each group (12 bits) to decimal numbers, then decrypt it by using a public key of the sender and a private key of the receiver presented in algorithm(1).

Algorithm (1) : Convert each number to 12 bits

```
1-  I = 1→ length of a secret message ( nmtxt)
2-  S = " ;  B = " ; C = nmtxt_i
3-  If ( C mod 2 ) = 0 ) then S = S + '0'
4-  If ( C mod 2 ) = 1 ) then S = S + '1'
5-  C = C div 2
```

6- If (C >0) then go to 3
7- J =1 → (12 – (length of (S))) S = S + '0'
8- J =12 → 1 B = B + S_j
9- Cph_i = B
10- Go to 1
11- End

- Convert each number after decryption to a corresponding letter until extracting all digital signatures.
- Extract the cipher text bits and convert each group (12 bits) to a decimal number then, decrypt it by using a private key of the RSA**[10]**.
- Convert each number after decryption to corresponding letter until extracting all plain text. This step will be described in detail.

5. Print the secret message and the digital signature.

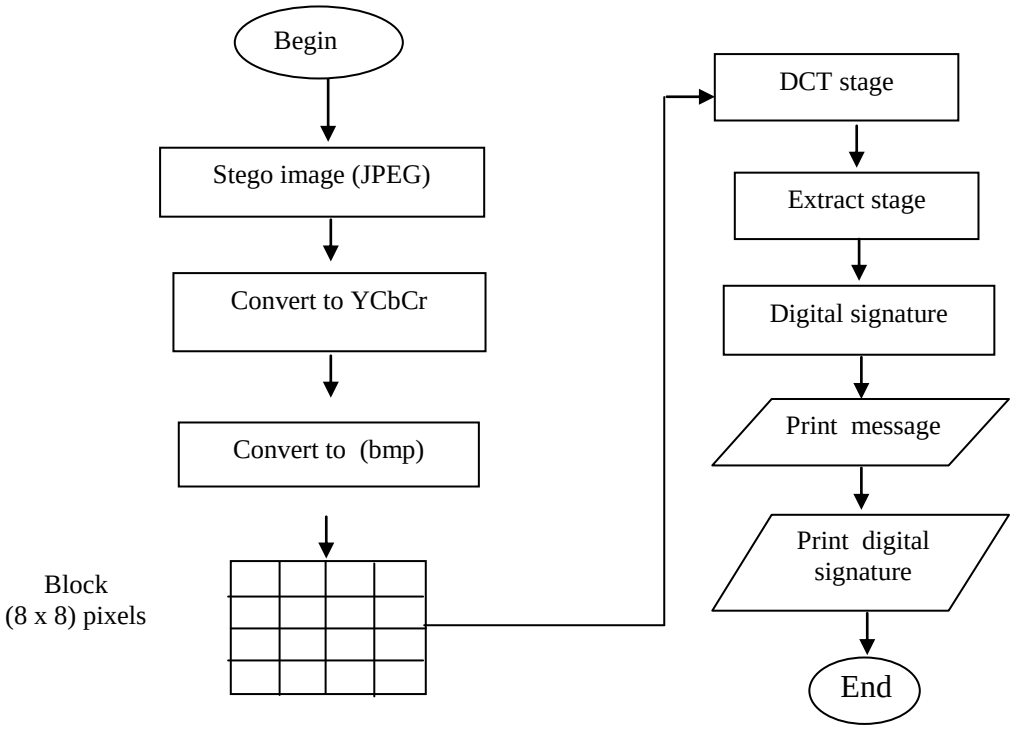


Figure (5): block diagram of decoding stage

2.2.4 *Extracting Bits*

1. In this stage, the bits are extracted from the stego-image. After converting the stego-image from JPEG to BMP, convert the image form RGB to YCbCr, separate the image components into blocks, each one consists of (8 x 8) pixels and transform each block (8 x 8) pixels to spatial frequency domain via the forward DCT. Figure (4) shows the block diagram for extracting bits from the stego-image. Some steps are implemented to extract the bits :
2. From each block the same DCT coefficients are used in

embedding stage choice to extract the bit. This coefficient is in position (0, 0) in each block.

3. Divide the value in position (0,0) by 16 and round the result called (**dc**) and inspect the result.

$$dc = round(block(0,0)/16)$$

   a. If the **dc** value equals odd numbers, this means one is hidden in it.
   b. If the **dc** value equals even numbers, this means zero is hidden in it.

4. Convert each 12 bits to decimal numbers.
5. Decrypt the decimal number by using the receiver key of the digital signature. Convert the result to a corresponding letter until finding (.) that refers to end the digital signature.
6. After finding all the characters of the digital signature, decrypt the decimal number by a private key (RSA algorithm) and convert the number after the decryption stage to a corresponding letter.
7. Steps one, two , three, four and five continue until finding the (#) character that refers to the end of the text.

Where Text : string contains the plain text, Sign : string contains the digital signature.

dc : real value refers to the hiding bit, TX : string [12] bits (obtain the extract bit from each block).

Nm : integer number, Ch : character.

3. *System Implementation*

The goal of this system is to embed the text and the digital signature in a cover-image (BMP format) to produce the stego-image (JPEG format).

System implementation accepts six inputs in the embedding stage:

- Input the file (BMP format), cover-image.
- Input the secret message.
- Input the public key to encrypt the text.
- Input the digital signature.
- Input the public key to encrypt the digital signature.
- Output file (JPEG format), stego-image.

System implementation accepts three inputs in the extracting stage:

- Input the file (BMP format),which contains the secret message.
- Input the privet key to decrypt the text.
- Input the privet key to decrypt the digital signature.

♦ *Embedding Steps*

The first of the proposed system, is embedding the message. This choice will lead to five main choices (Load Image BMP, Digital Signature, RSA Algorithm, Embedding Stage and Save Image JPEG), figure (6) shows the choice of the Embedding Steps, figure (7) illustrated the Digital Signature.

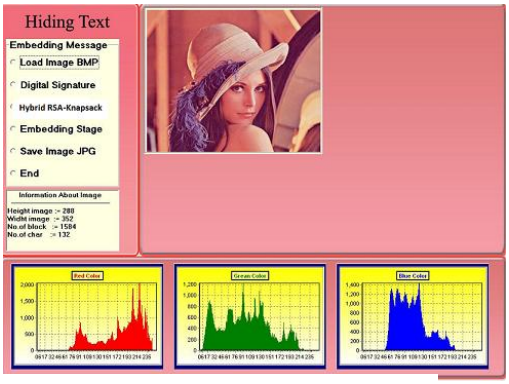


Figure (6): Cover-Image

Figure (7): Digital Signature

3.1 *Experimental Results*

In these experimental, the secret message, the digital signature and encrypting the message are used via the RSA algorithm, a second method of communication, called Steganography offers data protection in a somewhat different manner #. In Table (2) and Table (3) , the covers is Barrow, Lion images of size 352*288 respectively and system that has been used to embed the message contains (109) characters, and the digital signature contains (5) characters. Table (4) shows comparison between 2-LSB and proposed hiding DCT method.

Table (2): Rresults of embedded differential length text

Colour	PSNR	MSE	Text length	Compression ratio	Hiding time
Red	29.94	8.11	109 character	10	00:00:04
Green	30.77	7.37			
Blue	29.24	8.79			

Table (3): Results of the embedded differential length text

Colour	PSNR	MSE	Text length	Compression ratio	Hiding time
Red	33.64	5.3	93 character	14	00:00:04
Green	33.94	5.12			
Blue	32.35	6.15			

Table (4): Comparison between 2-LSB and proposed hiding DCT method

	2-LSB	DCT
Hiding time	Very short	Depends on image size
Security	Weak	Strong
Detection	Suspicious	Very difficult to be suspected
Size	Large	Small
Time	Very slow	Very fast

4.Conclusions

The proposed system provides the JPEG method with the digital signature and hybrid RSA with Knapsack cipher and hopes for an embedding text in an image. A number of conclusions are derived from this study:-

1. Steganography is an effective way to obscure data and hide sensitive information. The effectiveness of Steganography is amplified by combining it with cryptography. By using the properties of the DCT-LSB Steganography algorithm for image file and combining it with the hybrid RSA-Knapsack cryptography, we developed a method, which adds layers of security to the communication. Steganographic methods do not intended to replace cryptography but supplement it.
2. Steganography is not intended to replace cryptography but rather to supplement it. If a message is encrypted and hidden with a steganographic method it will provide an additional layer of protection and reduce the chance of the hidden message being detected.
3. The proposed system can be defined as asymmetric key Steganography since it uses two keys: a secret key and a public key between the sender and the receiver, in this system there is no need for the knowledge of the original cover in the extraction process.
4. In this system, we prove that if you hide information inside an image file (BMP) and that file is converted to another image format(JPEG), the hidden information will not be lost.
5. LSB in BMP is most suitable for applications where the focus is on the amount of information to be transmitted and not on the secrecy of that information, because LSB in BMP images are surely the suspicious ones that might arise from a very large BMP image being transmitted between parties. So we use JPG, because it is suitable for images that have to be communicated over an open system environment like the Internet.
6. From the implementation we conclude that the proposed system is very rapid in performing the extraction process and the size of the embedded text does not affect the speed of the system very much.

References

[1] Amit A. , Sherish J. "An Adaptive Steganography Technique for Gray and Colored Images" *IJARCSSE,* Volume 2, Issue 5, May 2012.

[2] Sharone G., "Combination of Cryptography and Steganography for Secure Communication Video File" California State Unversity, Sacramento, 2009.

[3] A.Nag, S. Biswas, D.Sarkar, P. P. Sakar," A noval Technique for Image Steganography based on Block_DCT and Huffman Encoding",Information Jornal of Computer Scince and Information Technology, Volume 2, Number 3,2010

.

[4] Alawy S., "Robust Information Hiding Techniques Using JPEG" University of Almusstnsry, Ms.c thesis in computer Science, 2004.

[5] Bushra Kassim Al-Abudi, "Colour Image Data Compression Using Multilevel Block Trunsaction Coding Technique*"*, Ph.d Thesis, College of Science, University of Baghdad, pp:20-21, 2002.

[6] Sua'd K.. Ahmad, *"*Image in Image Hiding System using Iterated Function System (IFS)*"*, Ms.c Thises, University of Sulaimani, pp:30, 2009.

[7] Shih T. Y. & Liu J.K., "On the Performance of JPEG2000 for Aerial Photo Compression", Department of Civil Engineering National Chiao-Tung University, 2001.

[8] Nada A. A. Mustafa," Design and Implementation proposed Encoding and Hiding Text in an Image", University of Sulaimani, Ms. c Thesis, pp:68-69, 2010.

[9] Trappe W. & Washington L., " Introduction to cryptography with coding theory". New Jersey: Prentice Hall, Rivest R. MD5.1992. Algorithm [Online]. Available: http://www.kleinschmidt.com/edi/md5.htm.

[10] William S. ," Cryptography and Network Security Principles and Practice", Fifth Edition, United states of America Prentice Hall, pp:267-277, 2011.

Fadhil Salman Abed is a Assistant Professor at the Depratement of Computer Sciences, Polytechnic University-Sulaimni Technical Institute of Kalar.. He received the B.Sc. degree in Mathematic from the University of Basra, Iraq, in 1987. He obtained his M.Sc. in Applied Mathematic(Computer Security) from University of Technology in 1997 and Ph.D. degree in Applied Mathematic(Fractal Image Compression) from University of Technology in 2004 . His research interests are in the field of Cryptography, Image Processing, Network security. He has many research papers in Image Processing and computer security.