ISSN: 0067-2904

Hiding and Encryption of Secret Image Using Secret Sharing Scheme

Yossra Hussain Ali, Hussein Jaeiz Mankhi*
Department of Computer Science, University of Technology, Baghdad, Iraq

Abstract

With the development of information technology and means for information transfer it has become necessary to protect sensitive information. The current research presents a method to protect secret colored images which includes three phases: The first phase calculates hash value using one of hash functions to ensure that no tampering with or updating the contents of the secret image. The second phase is encrypting image and embedding it randomly into appropriate cover image using Random Least Significant Bit (RLSB) technique. Random hiding provides protection of information embedded inside cover image for inability to predict the hiding positions, as well as the difficult of determining the concealment positions through the analysis of image or statistical analysis. Finally, secret sharing of carrier information image is applied. In the proposed hide method, the secret image is restored completely without external influence, where when image embedding is exposure to noise (95% - 90%) is retrieved of secret data and correlation coefficient is strong between the secret and retrieved image ranging from (0.97 to 0.93). Test results of the proposed method to measure the quality of image after reconstructing stego image from share images good. Where it has been hidden secret image (84x84) pixels into cover image (160x160) pixel, PSNR the result is 45.735db.

Keywords: Information Hiding, Message Digest Algorithm 5 (MD5), Secret Sharing

اخفاء وتشفير الصورة السرية باستخدام المشاركة السرية

يسرى حسين علي ، حسين جعيز منخي*
قسم علوم الحاسبات،الجامعة التكنولوجية ، بغدا د ، العراق

الخلاصة

مع تطور تكنولوجيا المعلومات ووسائل النقل للمعلومات أصبح من الضروري حماية المعلومات الحساسة. البحث الحالي يقدم طريقة لحماية الصور الملونة السرية يتضمن ثلاث مراحل: المرحلة الأولى حساب قيمة الهاش باستخدام احدى دوال الهاش لضمان عدم العبث او التحديث لمحتويات الصورة السرية. المرحلة الثانية هو تشفير الصور واخفاؤها عشوائيا داخل غطاء صورة مناسب باستخدام تقنية البت الأقل أهمية العشوائي (RLSB) الاخفاءالعشوائي يوفر حماية للمعلومات المتضمنة داخل غطاء الصورة لعدم امكانية التنبؤ بمواقع الاخفاء وكذلك من لصعوبة تحديد مواقع الاخفاء من خلال تحليل الصورة او التحليل الاحصائي. وأخيرا يتم تطبيق المشاركة

*Email: hussin_jaeiz@yahoo.com

السرية للصورة الحاملة للمعلومات. في طريقة الاخفاء المقترحة تم إخفاء واسترجاع الصورة السرية بشكل كامل بدون تأثيرات خارجية، حيث عند تعرض الصورة المضمنة للمعلومات الى ضوضاء تم استرجاع (95% – 90%) من البيانات السرية ومعامل الارتباط جدا قوي بين الصورة السرية والمسترجعة حيث يتراوح بين (0.97 الى 0.93). نتائج الاختبار للطريقة المقترحة لقياس جودة الصورة بعد إعادة تركيب Stego Image من الـ (share images) جيدة. حيث تم اخفاء صورة ملونة ابعادها (84x84) بكسل داخل غطاء صورة ابعادها (160x160) بكسل وكانت نتيجة الـ PSNRهو 45.735 ديسبل.

1. Introduction

Information hiding has received much attention in recent years as sharing of sensitive information via a common communication channel has become necessary [1]. Steganography comes from the Greek word steganos, which means concealed writing. It is an important research subject in the field of cryptography and information security. Steganography hides the secret message into a cover media to generate a stego-media, on which the existence of the embedded secret cannot be detected. Steganographic technique can overcome the conventional cryptographic approach, providing new solutions for secure data transmission without being suspected by censors. The cover media in a steganography scheme could be image, audio, video, document, etc. However, if the stego-media is lost or corrupted, the secret data cannot be reconstructed. Therefore, several secret sharing techniques have been proposed to overcome this weakness [2].

Secret sharing schema was introduced by Shamir in 1979 to provide confidentiality and security when secret visual information is transmitted through unsecured communication channels. Shamir suggested that the principle of secret sharing scheme, involves the secret D divided into a number of *n* pieces each piece called share or shadow and then distributed to a number of participants. In the case retrieval of the secret D requires just *k* or more participants to retrieve the secret image, where $k \leq n$ [3].

1. Related Work

Many of the methods and techniques developed in the field of information security include cryptography, steganography and secret sharing. There are many researches regarding secret sharing developed image sharing for certain purposes. Here the most important researches that touch on the secret sharing scheme are:

- **In 2002**, Thien et al [4] introduced a method to minimize the size of each share image to (1/k) of the original secret image.
- **In 2004**, Lin et al [5] proposed a method to share secret image into number of share images using secret sharing scheme, where each share image is hidden in image disguise selected by the user. Parity-bits is used to provide authentication and prevent participants from tampering or providing false images.
- **In 2007**, Yang et al [6] proposed secret image sharing with steganography and authentication, using the concept of hash function to provide the authentication for the purpose of improvement on the method Lin mentioned in [5].
- **In 2009**, C. Wu et al [7] proposed secret image sharing with steganography and authentication, this method is proposed to reduce the size and enlarge image camouflage used in [5,6].
- **In 2012**, Gupta et al [8] proposed an approach for hiding secret message in computer forensics including encoding the secret message in least significant bits (LSB) of cover image. Thereafter, the pixels of the image produced are updated using genetic algorithm, Then Stego image is considered as an input of secret sharing scheme.
- **In 2013**, Bidgar et al [9] proposed a method to encrypt secret image using a symmetric key produce stego image which is hidden in a number of images camouflage using secret sharing scheme to provide security of secret images against attackers.
- **In 2014**, Shruthi H R et al [10] proposed an approach to encrypt secret image by secret sharing scheme, then share images are compressed using discrete cosine transform (DCT) which is hidden in

random cover image using least significant bits (LSB). This approach increases the speed of transmission as well as giving additional security to secret image.

- **In 2015**, L. Jani Anbarasi et al [11] proposed an approach for encrypting secret images using DNA, then applying lossless secret sharing method and hiding share images in the host image. This approach achieves a better PSNR value without loss of secret pixel values.

2. Shamir's (k,n)Threshold Secret Sharing Scheme

Shamir's (k,n) threshold secret sharing is based on polynomial interpolation [12,13]. The details of this scheme are defined as follows:

1- The secret s is an integer number, n is the number of participants (the number of shares), the threshold k (k is No. of shares to retrieve the secret s), where $k \leq n$.

2- Defining $a_0 = d$, and choosing $(k-1)$ random number of the coefficients is $a_1, \cdots, a_{k-1}$, $where\ 0 \leq a_j \leq p-1$.

3- A prime number is chosen , where $p > \max(d,n)$. All further calculations are in the range $\{0, \cdots, p-1\}$ denoted by Z_p .

4- Using $(k-1)$ degree polynomial to compute the values of function $f(x_i)$, $where\ i = 1\ to\ n$, $x \in Z_p$

$$f(x_i) = \sum_{j=0}^{k-1} a_j\ x^j \qquad \cdots\cdots\cdots (1)$$

Then computing $s_i = f(x_i)\ mod\ p$, where $i = 1\ to\ n$ $\cdots\cdots\cdots (2)$

5- Delivering (x_i , s_i) as a share to n participants.

In order to reconstruct the secret, Lagrange interpolation formula will be used, which allows determining the polynomial f(x) of degree (k-1) from k shares (x_i, s_i), this formula is defined as follows [5]:

$$d = (-1)^{k-1}\left[f(x_1)\ \frac{x_2 x_3\ \cdots x_k}{(x_1-x_2)\,(x_1-x_3)\cdots(x_1-x_k)} + f(x_2)\ \frac{x_1 x_3\ \cdots x_k}{(x_2-x_1)\,(x_2-x_3)\cdots(x_2-x_k)} \ldots\ldots\ldots + f(x_k)\ \frac{x_1 x_2\ \cdots x_{k-1}}{(x_k-x_1)\,(x_k-x_2)\cdots(x_k-x_{k-1})}\right] mod\ p \quad \cdots\cdots\cdots\cdots\cdots\cdots (3)$$

In order to compute the values of coefficients a_i, where i=1 to k-1 the following formula will be used:

$$f(x) = \left[f(x_1)\ \frac{(x-x_2)(x-x_3)\ \cdots (x-x_k)}{(x_1-x_2)\,(x_1-x_3)\cdots(x_1-x_k)} + f(x_2)\ \frac{(x-x_1)(x-x_3)\ \cdots (x-x_k)}{(x_2-x_1)\,(x_2-x_3)\cdots(x_2-x_k)} \ldots\ldots\ldots + f(x_k)\ \frac{(x-x_1)(x-x_2)\ \cdots (x-x_{k-1})}{(x_k-x_1)\,(x_k-x_2)\cdots(x_k-x_{k-1})}\right] mod\ p \quad \cdots\cdots\cdots\cdots\cdots\cdots (4)$$

3. Hash Function

A hash function is a (mathematical) function which receives an input of arbitrary length and returns a function value (hash value) of a fixed length (usually 128 or 160 bits). Which are used in cryptography called one-way hash functions [14]. A hash function ***h*** is [15].

- **A one –way hash function:** It is not possible to generate a message from the hash value given.
- **A collision-resistant hash function:** It is not possible to generate two messages which have the same hash value. In general, the principal object of a hash function is data integrity.

4. Proposed Method

The proposed method is applied to hide digital colored images using the secret sharing. Figure-1 represents flowchart of the proposed method. The proposed method includes four phases:

1- Hash Function

Extraction hash value of the secret image is made by applying Message-Digest algorithm 5 (MD5). The main purpose of using the hash function is to ensure that the content of the secret image has not

been manipulated or tampered with its contents (add, delete, and update … etc.) by the unauthorized and malicious people, thereby maintaining the integrity of the secret image.

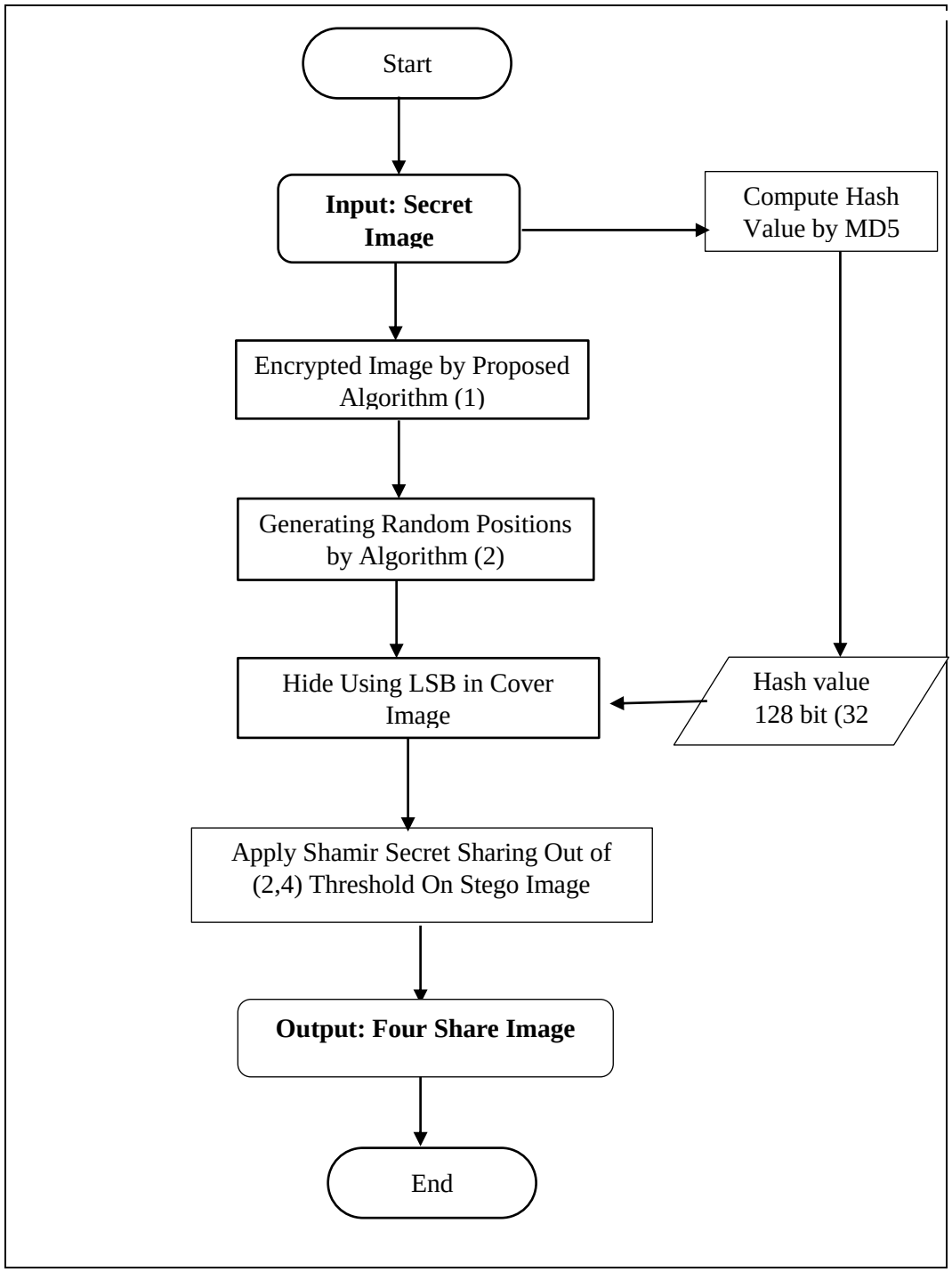


Figure 1-Flowchart of the proposed method

Encryption of Secret Image

The secret image is encrypted using proposed encryption algorithm. The proposed encryption algorithm is a block cipher; it encrypts plain image in 192 bit blocks. It can be described in five main steps:

1- The first step is the plain image is divided into blocks each one is 192 bits,

2- The second step is permutation using table called P_Box for each bit of block 192 bits. Permutation is done by changing position of bit to another without changing the value of the bit.

3- After permutation, mixing the resulting is made through a function called M_Function which divides each block output from the previous step to four sub block (A, B, C and D) the length for each one is 48 bits. Mixing function is applied through the logical functions as follows:

$$A1 = A \oplus B, B1 = A \oplus CLS(C)$$
$$C1 = CLS(A) \oplus D, D1 = CLS(A)$$

4- The fourth step includes ***nine*** rounds first round operates Exclusive or (Xor) between the key and the resulting value from the previous step and the subsequent ***eight*** rounds work Xor between the key and the resulting values for each round. Algorithm (2) describes method of generating keys.

5- The last step is doing a circular left shift (CLS) in each round thereby producing an encrypted image, these steps are described in Algorithm (1). The proposed encryption algorithm uses a single key to encryption and decryption, this means that the encryption key is from the type Symmetric ciphers.

Algorithm (1): Proposed Encryption of Secret Image
Input: Plain image, Secret Key 192
Output: Cipher image

Start
Step1: Divide image into blocks of 192 -bit
Step2: Permutation using table P_Box of block 192 -bit
Step3: for each block (192 bit)
3.1 Divide into four sub block (A, B, C and D)
3.2 Apply M_Function as follows
$A1 = A \oplus B, \quad B1 = A \oplus CLS(C)$
$C1 = CLS(A) \oplus D, \quad D1 = CLS(A)$
(A, B, C, D) = (B1, C1, D1, A1)
Step4: for r=1 to 9
4.1 Execute Exclusive or (Xor) between the block and key
(generated through Algorithm (2))
4.2 Circular Left Shift (CLS) of block
If r <= 9 then
go to step4
Else
go to step5
End if
End for
Step5: Repeat step1- step5 until all blocks are processed
End

Table P_Box

83, 150, 24, 91, 158, 32, 99, 166, 40, 107, 174, 48, 115, 182, 56, 123, 190, 64, 131, 5, 72, 139, 13, 80, 147, 21, 88, 155, 29, 96, 163, 37, 104, 171, 45, 112, 179, 53, 120, 187, 61, 128, 2, 69, 136, 10, 77, 144, 18, 85, 152, 26, 93, 160, 34, 101, 168, 42, 109, 176, 50, 117, 184, 58, 125, 192, 66, 133, 7, 74, 141, 15, 82, 149, 23, 90, 157, 31, 98, 165, 39, 106, 173, 47, 114, 181, 55, 122, 189, 63, 130, 4, 71, 138, 12, 79, 146, 20, 87, 154, 28, 95, 162, 36, 103, 170, 44, 111, 178, 52, 119, 186, 60, 127, 1, 68, 135, 9, 76, 143, 17, 84, 151, 25, 92, 159, 33, 100, 167, 41, 108, 175, 49, 116, 183, 57, 124, 191, 65, 132, 6, 73, 140, 14, 81,

148, 22, 89, 156, 30, 97, 164, 38, 105, 172, 46, 113, 180, 54, 121, 188, 62, 129, 3, 70, 137, 11, 78, 145, 19, 86, 153, 27, 94, 161, 35, 102, 169, 43, 110, 177, 51, 118, 185, 59, 126, 16, 67, 134, 8, 75, 142,

Proposed Keys Generator

The proposed method to generate the keys produces 256-bit length key then compresses the key (produced during the two phases of the method generating key) using table P-Compression into 192 bit. The proposed keys generator includes several steps:

1- The length of input key is 192-bits to be converted to hexadecimal (length 48 hex) divide to six block each one 8 hexadecimals.
2- Registers (X, Y, Z and W) are used to compute and generate key length of 128-bit each register holds 8 value hexadecimals, these registers are initialized to the following values in hexadecimal.
   X = "ac08d947", Y = "82a6fe5b", Z = "bd06e79f", W = "c8f5ee37"
3- This step consists of important two phases to generate a key length of 256-bit; each phase generates 128-bits involving four rounds, where each round consists of six steps.
4- The key generated from the previous step undergoes compression and permutation using a table called P-compression to result in length key of 192 bits, these steps are illustrated in Algorithm (2).

Algorithm (2): Keys generator
Input: Secret key (192 bit), four registers (X, Y, Z && W), No. of keys
Output: Set of secret keys, length of each one is 192 bits
Start **Step1**: Convert secret key (192 bit) into 48 hex **Step2**: Divide 48 hex into six times 8 Hex and store result into array_hex **Step3**: For i=1 to 2* No. of keys For round=1 to 4 For step=1 to 6 Key(i)= register (Y) + (CLR (array_hex(step) + register (X))) Update the values of registers (X, Y, Z, W) = (W, X, Y, Z) End for End for End for **Step4**: Compress the key length 256 bit into 192 bit using table P-Compression **End**

Table P_Compression

142, 199, 256, 56, 113, 170, 227, 27, 84, 141, 198, 255, 55, 112, 169, 226, 26, 83, 140, 197, 254, 54, 111, 168, 225, 25, 82, 139, 196, 253, 53, 110, 167, 224, 24, 81, 138, 195, 252, 52, 109, 166, 223, 23, 80, 137, 194, 251, 51, 108, 165, 222, 22, 79, 136, 193, 250, 50, 107, 164, 221, 21, 78, 135, 192, 249, 49, 106, 163, 220, 20, 77, 134, 191, 248, 48, 105, 162, 219, 19, 76, 133, 190, 247, 47, 104, 161, 218, 18, 75, 132, 189, 246, 46, 103, 160, 217, 17, 74, 131, 188, 245, 45, 102, 159, 216, 16, 73, 130, 187, 244, 44, 101, 158, 215, 15, 72, 129, 186, 243, 43, 100, 157, 214, 14, 71, 128, 185, 242, 42, 99, 156, 213, 13, 70, 127, 184, 241, 41, 98, 155, 212, 12, 69, 126, 183, 240, 40, 97, 154, 211, 11, 68, 125, 182, 239, 39, 96, 153, 210, 10, 67, 124, 181, 238, 38, 95, 152, 209, 9, 66, 123, 180, 237, 37, 94, 151, 208, 8, 65, 122, 179, 236, 36, 93, 150, 207, 7, 64, 121, 178, 235.

2- The Proposed Hiding Encrypted Image

The encrypted image resulting from the previous phase and hash value are concealed inside a suitable cover image using a Random Least Significant Bit (RLSB) technique, where it is generating random positions of cover image pixels through the entering of the first position by the user. The purpose of the random hiding is to provide protection of the information embedding in an image. where 8 bits per pixel are hidden (Each red and green color hides 3 bits and blue color hides 2 bits). Algorithm (3) describes method of generating random positions of cover image pixels.

Algorithm (3) Generating random positions of cover image pixels

Input: Cover image, No. of pixels required , First position (for the X && Y Axises), X is width of cover image; Y is height of cover image, h=X, flag = 0

Output: Array Random Points of Axis of y () and Axis of x ()

Start
Step1: Generate initialize point (R, C) by user
Step2: For i=1 To (No. of pixels) – 1
Flag = 0
R = Axis of y (i-1),
C = Axis of x (i-1)
2.1. R_{New}= (C * h) – (Axis of Y (i-1)) Mod X
C_{New}= (R* i) – (Axis of X (i-1)) Mod Y
2.2. If R_{New} or C_{New} is smaller than zero, then
Update the value of R_{New} or C_{New}
2.3. If R_{new} is not found in Axis of x or C_{New} is not found in Axis of y, then
Axis of x (i) = R_{New},
Axis of y (i) = C_{New}
Else
R=1, h=h-1
flag=flag+1
if flag>=2 then R = (X*Y) - flag
Goto **2.1**
End If
Step3: Repeat Step2 until all No. of the required pixels are processed
End

3- ***Secret Sharing***

The concept of the secret sharing is applied to Stego image resulting from the previous phase. Shamir (2,4) threshold secret sharing scheme is used to generate four shares of stego image. The secret image sharing involves taking two pixels of stego image and select all coefficients a_{k-1}-degree polynomial to replace them with two value pixels, instead of selected random coefficients as used in the method of Shamir. The result obtained from the polynomial equation is substituted using table called S_Box. This method reduces the size of the image share to 1/k of stego image. Algorithm (4) explains a method for applying secret image sharing.

Algorithm(4): Shamir's (2,4) Threshold secret image sharing

Input: Stego image, n=4, k=2, p=251

Output: Four Shares Image

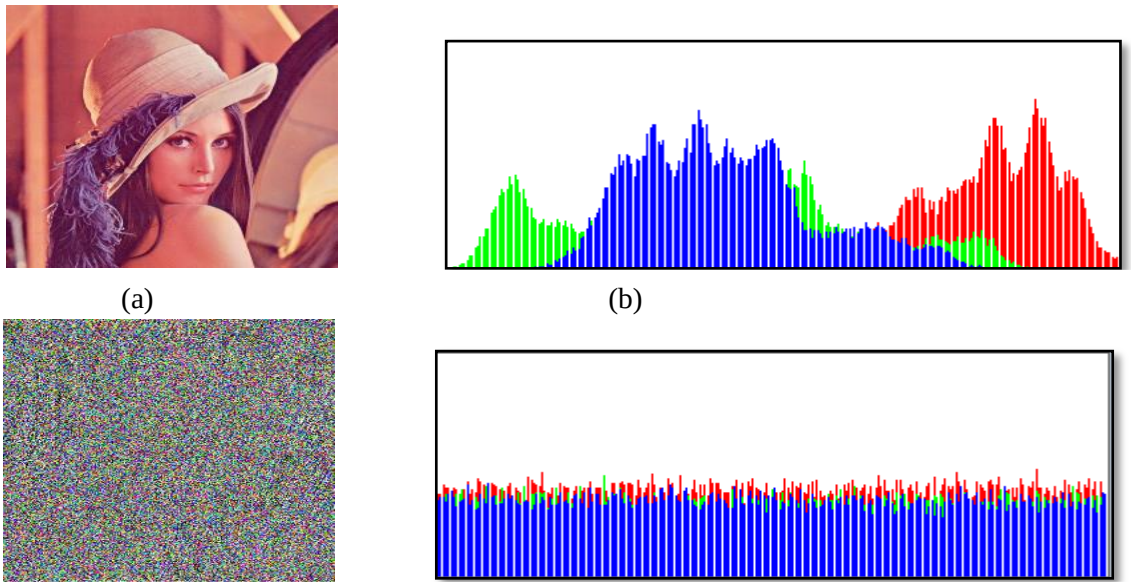
Start:
Step1: for w=0 to width_stego image
for h=0 to height_stego image
Begin
Read two pixels of stego image // R, G, B
Step2: For $x = 1\ to\ n$
2.1. $f(x) = \left(\sum_{j=0}^{k-1} a_j\, x^j\right) mod\ p$
2.2. Share(x)= S_Box(f(x))
2.3. Deliver the result into share $image_x$ (x, Share(x))
End for
Step3: Repeat step1 and step2 until all image pixels are processed
End

Table S_Box

99 , 124 , 119 , 123 , 242 , 107 , 111 , 197 , 48 , 1 , 103 , 43 , 187 , 215 , 171 , 118 , 202 , 130 , 201 , 125 , 250 , 89 , 71 , 240 , 173 , 212 , 162 , 175 , 156 , 164 , 114 , 192 , 183 , 84 , 147 , 38 , 54 , 63 , 247 , 204 , 52 , 165 , 229 , 241 , 113 , 216 , 49 , 21 , 4 , 199 , 35 , 195 , 24 , 150 , 5 , 154 , 7 , 18 , 128 , 226 , 235 , 39 , 178 , 117 , 9 , 131 , 44 , 26 , 27 , 110 , 90 , 160 , 82 , 59 , 214 , 179 , 41 , 227 , 47 , 132 , 83 , 209 , 0 , 237 , 32 , 176 , 177 , 91 , 106 , 203 , 190 , 57 , 74 , 76 , 88 , 207 , 208 , 239 , 170 , 15 , 67 , 77 , 51 , 133 , 69 , 249 , 2 , 127 , 80 , 60 , 159 , 168 , 81 , 163 , 64 , 143 , 146 , 157 , 56 , 245 , 188 , 182 , 218 , 33 , 16 , 22 , 243 , 210 , 205 , 12 , 19 , 236 , 95 , 151 , 68 , 23 , 196 , 167 , 126 , 61 , 100 , 93 , 25 , 115 , 96 , 129 , 79 , 220 , 34 , 42 , 144 , 136 , 70 , 238 , 184 , 20 , 222 , 94 , 11 , 219 , 224 , 50 , 58 , 10 , 73 , 6 , 36 , 92 , 194 , 211 , 172 , 98 , 145 , 149 , 228 , 121 , 231 , 200 , 55 , 109 , 141 , 213 , 78 , 169 , 108 , 86 , 244 , 234 , 101 , 122 , 174 , 8 , 186 , 120 , 37 , 46 , 28 , 166 , 180 , 198 , 232 , 221 , 116 , 31 , 75 , 189 , 139 , 138 , 112 , 62 , 181 , 102 , 72 , 3 , 246 , 14 , 97 , 53 , 87 , 185 , 134 , 193 , 29 , 158 , 225 , 248 , 152 , 17 , 105 , 217 , 142 , 148 , 155 , 30 , 135 , 233 , 206 , 85 , 40 , 223 , 140 , 161 , 137 , 13 , 191 , 230 , 66 , 104 , 65 , 153 , 45

5. Experimental Results

This section presents implementation and experimental results of the proposed method in the hiding secret image using a secret sharing, where several of tests are applied to evaluate the performance of the proposed method. A test procedure is based on image quality measures between cover and stego image, in addition encryption quality of images. Figure-3 shows the histograms of cipher image generated by proposed encryption algorithm and the corresponding plain image. Table-1 illustrates entropy and correlation coefficients between the original and encrypted imag



(a) (b)

(c) (d)

Figure 3- The encryption results, (a) Original image (250x250) (b) Histogram of original image (c) Encrypted image (d) Histogram of encrypted Image.

Table 1- Entropy and correlation coefficient between the original images and their corresponding encrypted images

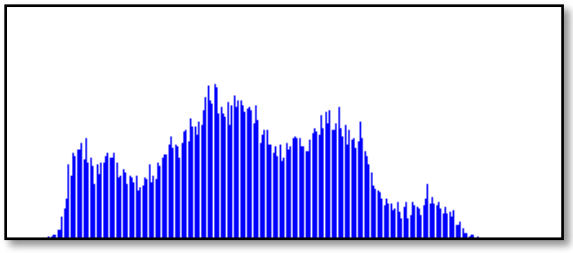
Measures / Test Images	Entropy for Plain Image	Entropy for Cipher Image	Correlation Gray	Correlation Red	Correlation Green	Correlation Blue
Lena	**7.3109**	**7.4642**	**0.00602**	**0.00300**	**0.00623**	**0.00411**

Baboon	**7.2202**	**7.4538**	**0.00894**	**0.00310**	**0.00493**	**0.00726**
Jet-plane	**6.6356**	**7.4735**	**0.00182**	**0.01803**	**0.00405**	**0.00879**
Lichtenstein	**7.3706**	**7.4816**	**0.00877**	**0.01426**	**0.00684**	**0.00041**
Pepper	**7.3811**	**7.4717**	**0.00968**	**0.00051**	**0.00499**	**0.00657**
Barbara	**7.4468**	**7.4539**	**0.00197**	**0.00023**	**0.00457**	**0.00169**

Secret image (84x84)

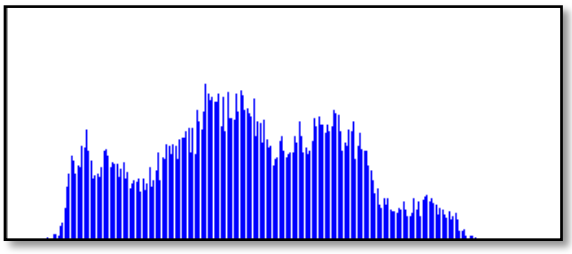
Encrypted Secret image

Cover Image (160x160)



Histogram of Cover Image

Stego Image (160x160)



Histogram of Stego Image

Share Image 1

Share Image 2

Share Image 3

Share Image 4

Figure 4- Explains hiding encrypted secret image randomly inside the cover image then applying secret sharing scheme of (2,4) threshold

Table 2- Illustrates the results of quality measures of cover and stego images

Measures / Test images	Dimension Cover Image	Dimension Secret Image	MSE	PSNR
Lena	160x160	84x84	1.562	45.735
	250x250	128x128	1.512	45.875
Baboon	160x160	84x84	1.521	45.850
	250x250	128x128	1.472	45.991
Barbara	160x160	84x84	1.502	45.903
	250x250	128x128	1.471	45.994

Table-3 explain evaluate the image quality of the stego-image (when camouflage images are used as cover image for share images) among Lin–Tsai's method, Yang *et al.'s* method, C. Wu *et al.'s* method, and the proposed method. The mean square error (MSE) and the peak signal-to-noise ratio (PSNR) are used to compare image quality. two secret images are (House, Clock) with 256x256 pixels. Four cover images are (Airplane, Baboon, Lena, Pepper) with 512x512 pixels

Table 3- The image quality comparisons in (2,4) threshold among several methods and our method

Secret Image	Stego-Image	Lin-Tasi's method		Yang *et al.'s* method		Wu *et al.'s* method		Our Method	
		PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE
House	Airplane	39.21	7.80	41.58	4.52	41.96	4.14	47.74	0.98
	Baboon	39.17	7.86	41.52	4.58	41.96	4.15	47.63	1.04
	Lena	39.19	7.84	41.53	4.58	41.95	4.15	47.53	1.02
	Pepper	39.17	7.87	41.51	4.60	41.94	4.16	47.64	1.02
Clock	Airplane	39.21	7.79	41.62	4.48	41.95	4.15	47.74	0.98
	Baboon	39.19	7.85	41.55	4.55	41.95	4.15	47.63	1.05
	Lena	39.19	7.85	41.56	4.54	41.95	4.15	47.54	1.02
	Pepper	39.19	7.87	41.55	4.56	41.97	4.14	47.65	0.99

6. Conclusion

The proposed method includes three levels of protection. First level, encrypts secret image using the proposed algorithm, then hides encrypted image randomly into an appropriate cover image. Finally, the concept of secret sharing of stego image is applied to ***n*** of share images depending on Shamir method. It accomplishes ability to find out change or tamper in the secret image using a hash function. Test results of the proposed method are based on the standards used to measure image quality after hiding, as well as the encrypted image quality is good.

References

1- Sumathi. C. P, Santanam, T and Umamaheswari. G. **2013**. A Study of Various Steganographic Techniques Used for Information Hiding. *International Journal of Computer Science & Engineering Survey (IJCSES)*, 4(6).

2- Yang, C. N, Ouyang, J.F. and Harn, L. **2012**. Steganography and authentication in image sharing without parity bits. *Optics Communications*, 285, pp: 1725–1735.

3- Salehi, S. and Balafar, M. A. **2015**. An Investigation on Image Secret Sharing. *International Journal of Security and Its Applications*. 9(3), pp:163-190.

4- Thien, C. C. and Lin, J. C. **2002**. Secret Image Sharing. Elsevier, *Computers & Graphics 26*, pp: 765–770.

5- Lin, C. C. and Tsai, W. H. **2004**. Secret image sharing with steganography and authentication. Elsevier, *The Journal of Systems and Software 73*, pp: 405–414.

6- Yang, C. N, Chen, T. S., Yu, K. H. and Wang, C. C. **2007**. Improvements of Image Sharing with Steganography and Authentication. *Journal of Systems and Software*, 80(7), pp: 1070-1076.

7- Wu.C.C, Hwang.M.S and Kao.S.J. **2009**. A new approach to the secret image sharing with steganography and authentication. *The Imaging Science Journal*, 57(3).

8- Gupta, R. Jain, A. and Singh, G. **2012**. Combine use of Steganography and Visual Cryptography for Secured Data hiding in Computer Forensics. *International Journal of Computer Science and Information Technologies*, 3 (3), pp: 4366 – 4370.

9- Bidgar, P. and Shahare, N. **2013**. Key based Visual Cryptography Scheme using Novel Secret Sharing Technique with Steganography. *Journal of Electronics and Communication Engineering (IOSR- ECE)*,8(2), pp:11-18.

10- Shruthi, H. R. Ranjan, K. H. and Prasanna, K. H. **2014**. A Visual Secret Sharing Technique for Secure and Fast Transmission of Image. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(4).

11- Anbarasi, L. J. Mala, G. S. and Narendra, M. **2015**. DNA based Multi-Secret Image Sharing. International Conference on Information and Communication Technologies, Procedia Computer Science. 46, pp: 1794 – 1801.

12- Shamir, A. **1979**. How to Share a Secret. *Communications of the ACM*, 22(11), pp:612–613.

13- Ghazi, M. **2013**. Authenticated Image Documents Using Secret Sharing Technique. M.Sc. Thesis. Department of Computer Science, University of Technology, Baghdad, Iraq.

14- Stallings, W. **2011**. *Cryptography and Network Security*: Principles and Practice, Fifth Edition, Prentice Hall.

15- Konheim, A. G. **2007**. *Computer Security and Cryptography*, Wiley.