Volume 5, No. 6, June 2014
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

A NEW STEGANOGRAPHY DATAHIDING ALGORITHM

[1]Hussain Abo Surrah, [2]Isbudeen Noor Mohamed
College of Computers and Information Technology, Taif University, KSA.
[1]salama366@yahoo.com, [2]isbudeen@hotmail.com

Abstract- Steganography is the art or practice of concealing a message, image, or file within another message, image, or file. It serves as a better method of securing message than cryptography which only conceals the content of the message not the existence of the message. This means that, the original message is being hidden within a carrier such that the changes so occurred in the carrier are not observable. In this paper, we proposed a new steganography scheme for hiding a piece of critical information in a host binary image. The proposed scheme presents a new form of weight matrix and a secret key. The weight matrix is dynamic that is it changes its position at every block of the image to increase the security. The performance of the proposed scheme is compared with that of the Chen et al scheme (CPT scheme). The results show that our scheme provides a higher security, and hides a higher amount of data than CPT scheme, and it keeps that same quality level of the host image as the CPT scheme.

Keywords- Binary image processing, Datahiding, PSNR, Steganography

INTRODUCTION

As digital media became easily accessed, copied, multiplied without information loss, and manipulate it without detection, their security-related issues become a greater concern. One main issue is confidentiality, which is typically achieved by encryption. Ciphering doesn't allow any modification on any image or audio content. It is unsuitable for verifying the ownership of images and audio[2]. Information hiding is the process of inserting secret data into digital media by modifying original multimedia content to distract opponent's attention [1,2,3]. One less confusing name for data hiding is the steganography. Steganography would be combined with encryption to achieve a higher level of security. The application of data hiding can be used in military, commercial, and anti-criminal-related and so forth [6,7]. Classification of information hiding techniques can be found in [8,4,5]. The most common used technique for data hiding in the gray images is the least significant bit (LSB)[12]. A genetic algorithm is proposed in [13]to degrade the quality of the image. A hiding scheme base donkey stream generator is proposed in [6,12]. It considers show to apply the public key cryptography to steganography. On the other hand, very little research had been done for data hiding in binary images, since each pixel in binary images requires only one bit to represent it, and the changing in any pixel can be easily detected [9,10,11].

The CPT scheme [14] proposed a Secure data hiding algorithm that cannot only embed a great deal of secret information into binary images but also ha imperceptible quality. In that scheme the authors proposed a secret key, and a weight matrix to protect the secret data and increased the capacity of data that can be hidden in an image block of size $m\times n$ in to $\log_2[m\times n+1]$.In this paper, we propose a modified steganography scheme for hiding apiece of critical information in a host binary image. A secret key and a weight matrix are used to protect the hidden data. Given an image block of size $m\times n$, the proposed scheme can hide as many as $\log_2[m\times n+1]+1$ bits of data in the host image by changing at most 2 bits. This scheme is compared to the CPT scheme [14]. From the comparison, it is easy to see that, the proposed scheme can provide higher security, embed more data than the CPT scheme, and maintain the same quality of the stego image.

This paper is organized as follows: Section 2, presents the description and the assumptions of CPT scheme. Section 3. Presents the proposed algorithm. Section 4., describes, analyzes and compares the results obtained using the proposed and the CPT algorithms. Finally, Section 5. Summarizes this paper.

REVIEWS AND MOTIVATIONS

In this section, we present the data hiding scheme which introduced by Chen et al [14], and is abbreviated as CPT scheme. A block diagram of this scheme is shown in Fig.1.
In the CPT scheme, given a host binary image F.F will be partitioned in to block suffixed size$m\times n$(for simplicity, assume that F's size is a multiple of $m\times n$). The scheme is able to hide as many as $r\leq\log_2[m\times n+1]$ bits of secret data in each host block by modifying at most 2 bits in the block. The secret key has two components:

a. K:is a randomly selected binary matrix of size $m\times n$
b. W: weight matrix which is an integer matrix of size $m\times n$. It satisfies equation(1):

$$\mathbf{W}=\{W_{i,j}|i=1\cdots m,j=1\cdots n\}=\{1,2,\cdots,2^r-1\}. \quad (1)$$

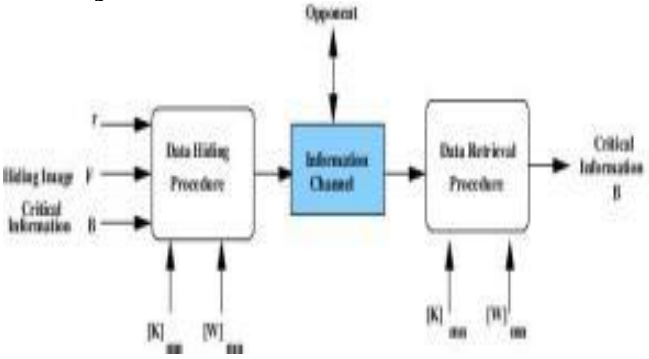


Figure1: block diagram of the CPT data hiding scheme

W can be calculated by first picking 2^r –entries from the $m \times n$ matrix and assigning $\{1,2,\cdots,2^r-1\}$ to these entries, in any order, the new assign any value to each of the remaining $m\times n-(2^r-1)$. The data hiding is achieved by modifying some bits of F. The process of hiding a bit stream $b_1 b_2 \cdots b_r$ into an $m\times n$ host block, say, F_i can be summarized as follows:

a. Compute $F_i \oplus K$, where $\oplus$ means the bit wise X-OR of two equal-size binary matrices.
b. Compute SUM $((F_i \oplus K) \otimes W)$, where $\otimes$ means the pair-wise multiplication of two equal- size matrices, and SUM means the sum of all elements in a matrix.
c. From the matrix $F_i \oplus K$, compute for each $w=1\cdots 2^r-1$the following set:

$S_w=\{(j,k)|(([W]_{j,k}=w)\wedge([F_i \oplus K]_{j,k}=0))\vee(([W]_{j,k}=2^r-w)\wedge([F_i \oplus K]_{j,k}=1))\}$(2)

d. Define a weight difference

$d\equiv(b_1 b_2\cdots b_r)-SUM((F_i \oplus K)\otimes W] \bmod 2^r$.

If $d=0$, there is no need to change F_i.Otherwise:

(a) Randomly pick $h\in\{0,1,2,\cdots,2^r-1\}$such that $S_{hd}\neq\emptyset$ and $S_{(h-1)d}\neq\emptyset$.

(b) Randomly pick $(j,k)\in S_{hd}$ and complement the bit $[F_i]_{j,k}$.

(c) Randomly pick $(j,k)\in S_{-(h-1)d}$ and complement the bit$[F_i]_{j,k}$.

e. On receiving $\grave{F}_i$, the receiver computes SUM $((\grave{F}_i \oplus K) \otimes W)$ mod2^r to find the hidden bit stream $b_1 b_2\cdots b_r$.

For more details about this process see [14]. As an example, let the host image F, secret key K, and weight matrix W be as shown in Fig.2.

First, F is partitioned in to two4 × 4blocks, say, F1 and F2. We can hide as r ≤ [log2 4 × 4 + 1] bit sin each block. Let r = 3and the secret data = 000001, (the first three bits will be embedded in F1 and the last three bits will be embedded in F2).

The results of F1 ⊕K, and F2 ⊕K are shown in Fig.3(a). To embed 000 in F1, since SUM ((F1 ⊕K) ⊗W) mod23 = 2 and the weight difference d = (0 − 2) mod 23 = 6, thus we want to increase the weight by 6, this will be done by complementing [F1]4,4. To embed 001in F2, since SUM ((F2 ⊕K) ⊗W) mod23 = 4 and the weight difference d = (1 − 4) mod 23 = 5, thus we want to increase the weight by 5. There is no single point in F2 by complementing which we can do so, hence changing the two bits of F2 is necessary. One possibility is S10 = S2 = {(2, 2)}, and S−5 = S3 = {(1, 3), (2, 1), (3, 2), (4, 3)}. In this example, we choose to complement [F2]2,2 and [F2]3,2. The final modified image is shown inFig.3(b).

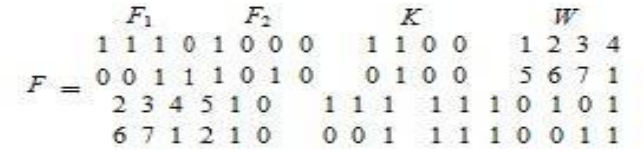


Figure2: An example of the host image F, secret key K, and weight matrix W

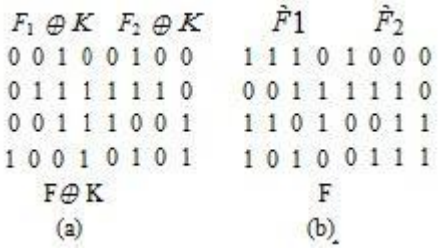


Figure 3: (a) *Fi* ⊕ *K* , and (b) the modified host image *F`i*

THEPROPOSEDALGORITHM

In this section, we present a new data hiding scheme.
The inputs to this scheme are:

a. F: a host bitmap image F which is partitioned in to block so size $m\times n$.
b. K: a secret key matrix of size $m\times n$ shared by the sender and the receiver.
c. W:a secret weight matrix shared by the sender and the receiver. It is an integer matrix of size $m\times n$. Its elements are from $\{2^{r-1}\cdots 2^r-1\}$ at any secret randomly order, where r is the number of embed de db it's in to $m\times n$ block of F.
d. S: a secret substitute weight transformation, for changing the position for the weight matrix in each hiding block.
e. B: some secret information consisting of $k\times r$ bits to be embedded in F, where k is the number of $m\times n$ block sin F.

Fig.4 shows a block diagram of the proposed data hiding scheme.

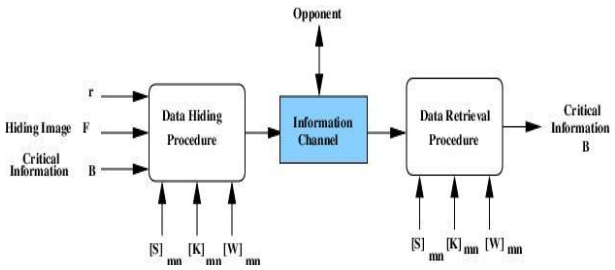


Figure 4: block diagram of the proposed data hiding scheme

Dynamic Weight Management:

The proposed scheme heavily relies on the dynamic weight matrix W to represent the embedded data, and on the substitution matrix S to change the position of the dynamic weight matrix in each embedding block. The following example illustrates how the proposed dynamic secret weight matrix works. Suppose that the size of K, the initial weight matrix W_0, and the secret substitution matrix S equal (3×3).Considera3×3firstblock$F_1$of the host image F as shown in Fig.5. We illustrate how the dynamic weight matrix S works in the proposed scheme to embed r=4 bits of

data into F_1, say this bits are $b_1b_2b_3b_4$. We First perform the bit wise $X-OR$ on F_1 and K, and substitute by W_0 in the substitution matrix S to get W_1, after that we compute $(F_1 \oplus K) \otimes S(w_0)$ asinFig.6.

F_1	K	W_0	S
1 1 0	0 1 1	8 9 10	$[W]_{23}$ $[W]_{31}$ $[W]_{11}$
1 0 0	1 1 0	11 12 13	$[W]_{22}$ $[W]_{12}$ $[W]_{21}$
1 1 0	0 1 1	14 15 8	$[W]_{13}$ $[W]_{33}$ $[W]_{32}$

Figure 5:A dynamic weight example: the inputs F, K, W_0, S

$F_1 \oplus K$		$w_1 = S(w_0)(F_1 \oplus K) * W_1$		
1 0 1		13 14 8	13 0 8	
0 1 0	*	12 9 11	= 0 9 0	
1 0 1		10 8 15	10 0 15	

Figure 6: The pair-wise multiplication operation between $(F_i \oplus K)$ and $S(w)$

For all the non zero elements obtained from $X-OR$ $((\grave{F}_i \oplus K) \otimes S(W_{i-1}))$,we compute

X-OR. Applying this step in our example we have $(13 \oplus 8 \oplus 9 \oplus 10 \oplus 15) = 1101 \oplus 1000 \oplus 1001 \oplus 1010 \oplus 1111 = 1001 = 9$.Hence, we will be able to embed 4 data bits $b_1b_2b_3b_4$, into F_i, suppose that F_i is changed to $\grave{F}_i$. Our scheme ensures the following invariant:

$$X-OR\ ((\grave{F}_i \oplus K) \otimes S\ (W_{i-1})) \equiv b_1b_2 \cdots b_r \quad (3)$$

With this invariant, the receiver can retrieve $b_1b_2b_3b_4$, by computing $X-OR$ $((\grave{F}_i \oplus K) \otimes S\ (W_{i-1}))$. How $\grave{F}_I$ is computed to ensure invariant (3)?

We intend to have as few bits in F_i as possible. Since $X-OR$ $((F_1 \oplus K) \otimes S\ (W_0)) \equiv 9$, if fortunately $b_1b_2b_3b_4 = 9$, then there is no need to modify F_i. Otherwise some bit (s) has to be modified. If we compute the difference between $b_1b_2b_3b_4$and$9 = 1001$(i.e. the $X-OR$ between them), we can determine what bits can be modified in $\grave{F}_i$, let the deference be$x$, we have the following two probabilities:

1.$x \geq 2^{r-1}$ (which is the small lest value of W), then by complementing the pixel in$[F_i]$,which is corresponding to the value of $x$in to $S\ (W_{i-1})$.

2.$x < 2^{r-1}$, then by complementing the two pixels in $[F_i]_{ij}$ which are corresponding o the-1).

Value of $x+2^{r-1}$, and the value 2^{r-1} in to $S(W_i$

As are sult, the value of $X-OR$ $((\grave{F}_i \oplus K) \otimes S\ (W_{i-1}))$ $\equiv b_1b_2b_3b_4$ is obtained.

Hidingsteps:

Definition 1 *(The weight matrix): An $m \times n$ matrix W can serve as a weight matrix if each element of $\{2^{r-1}, 2^{r-1}+1, \cdots, 2^r-1\}$ appears atleast once in W.*

Based ondefinition1, for all $m \times n = 3 \times 3$block size, we take the element so fW from the range2^{r-1} to 2^r-1. For instance if $m \times n = 4 \times 4$, then our scheme can embed r=5bits, where

$2^{r-1} = m \times n = 16$, so we can choose the weight elements form the set $\{16,17,\cdots,30,31\}$.

In the case of $m \times n = 3 \times 3$, seven element so fW takes the values from the range 2^2to2^3,and the remaining element can be selected randomly from this range.

Briefly we summarize the steps of imbedding r bits of data say $b_1b_2 \cdots b_r$ into $F_i$by changing
At most 2 bits in F_i as follows:

a. S_1*:* Compute $F_i \oplus K$.
b. S_2*:* Compute $W_i = S(W_{i-1})$ by using substitution box S.
c. S_3*:* Compute $(F_i \oplus K) \otimes W_i$.
d. S_4*:*For all nonzero elements in $(F_i \oplus K) \otimes W_i$ compute $\grave{S}_r = X-OR((F_i \oplus K) \otimes W_i)$.
e. S_5*:* Compare the secret data S_r by the $\grave{S}_r$, using the X-OR$(\grave{S}_r \oplus S_r)$.
f. S_6*:* Convert $(\grave{S}_r \oplus S_r)$ in to its integer number (say it x).
g. S_7*:* If $x \geq 2^{r-1}$, then Change the bit in F_i which is corresponding to the value $x$in W_i.

Else If $x < 2^{r-1}$, the nx is not in the weight matrix. In this case do the following:

a) Change the bit in F_i which is corresponding to the element2^{r-1} in the weight matrix.
b) Change the bit which is corresponding o the $2^{r-1} + x$ in the weight matrix
c) S_8*:*To retrieve the hidden data, insure equation (3).

Below, we demonstrate an example, let the host image F, and secret key K, initial weight matrix W area sin Fig.7, and the substitution matrix S is a sin Fig.8.

First, $F$is partitionedinto4×4blocks,sayF_1,F_2,F_3,and F_4,and compute $(F_i \oplus K) \otimes S(W_{i-1})$.
Calculations of $F_i \oplus K$, and $(F_i \oplus K) \otimes S(W_{i-1})$ are in Figures 9(a), and 9(b) respectively.

We can hide a s$r \leq [\log_2 4 \times 4 + 1] + 1$bits of data in each block.Let$r$=5, and the secret
S_r, where $S_r = S_{r1}S_{r2}S_{r3}S_{r4}$ =00010111110101000110
•To get S_{ri}, convert each element in $\omega = (F_i \oplus K) * W$ to binary and XOR them.
–$\grave{S}_{r1} = X-OR$ (20 $\oplus$21$\oplus$22$\oplus$25$\oplus$23$\oplus$17$\oplus$30)=10100 $\oplus$ 10101$\oplus$10110$\oplus$11001$\oplus$10111$\oplus$10001$\oplus$11110=10110.

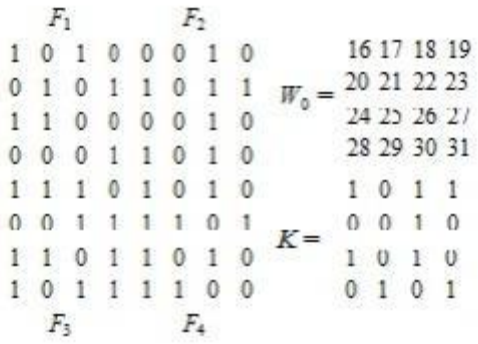


Figure 7: An example of host image *f*, secret key *K*, and weight matrix *W*.

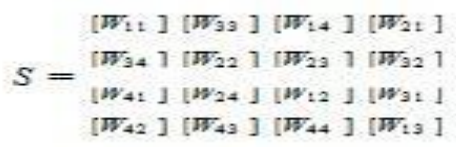


Figure 8: The substitution matrixs.

–
$\grave{S}_{r2}=X-OR(16\oplus27\oplus24\oplus23\oplus29\oplus30\oplus31\oplus18\oplus19)=10000\oplus$
$11011\oplus11000\oplus10111\oplus11101\oplus11110\oplus11111\oplus10010\oplus10011=11001$

–$\grave{S}_{r3}=X-OR(26\oplus24\oplus25\oplus23\oplus17\oplus29\oplus31\oplus19\oplus18)=11010\oplus$
$11000\oplus11001\oplus10111\oplus10001\oplus11101\oplus11111\oplus10011\oplus10010=11110$

–$\grave{S}_{r4}=X-OR(28\oplus29\oplus21\oplus22\oplus23\oplus18\oplus27)=11100\oplus11101\oplus$
$10101\oplus10110\oplus10111\oplus10010\oplus11011=11100.$

•Compute $\grave{S}_{ri}\oplus S_{ri}$, and then compare it with 2^r

–$\grave{S}_{r1}\oplus S_{r1}=10110\oplus00010=10100=20>16.$

–$\grave{S}_{r2}\oplus S_{r2}=11001\oplus11111=00110=6<16.$

–$\grave{S}_{r3}\oplus S_{r3}=11110\oplus01010=10100=20>16.$

–$\grave{S}_{r4}\oplus S_{r4}=11100\oplus00110=11010=26>16$

•Then change the pixel which is corresponding to the weight in $W_i$that has the same value

$\grave{S}_{ri}\oplus S_{ri}$

–in F_1 change the pixel which is corresponding to the weight 20 in W_1.

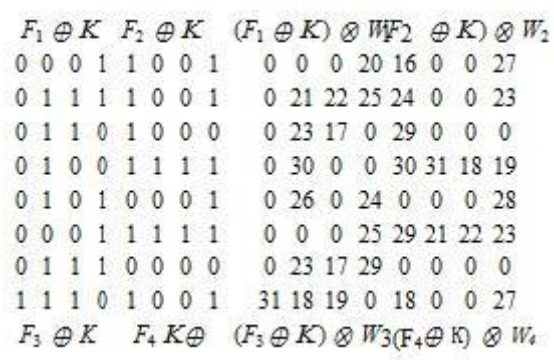


Figure 9:(a) $F_i\oplus K$, and (b) $(F_i\oplus K)\otimes W_{i-1}$

–in F_2 change the two pixels which are corresponding to the weight 16, and the weight
22(*i.e.*16+6) in W_2.

–in F_3 change the pixel which is corresponding to the weight 20 in W_3.

–in F_4 change pixel which is corresponding to the weight 26 in W_4.

•then $\grave{F}_I$ is obtained as showninFig.10

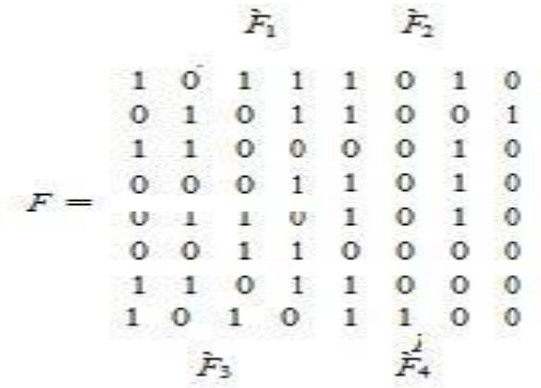


Figure 10: The stego image $\grave{F}$

•To retrieve data from $\grave{F}_I$ compute $(\grave{F}_i\oplus K)\otimes S\ (W_i-1)$

–S_{r1}
$=X-OR(21\oplus22\oplus25\oplus23\oplus17\oplus30)=10101\oplus10110\oplus11001\oplus$
$10111\oplus10001\oplus11110=00010.$

–S_{r2}
$=X-OR(27\oplus24\oplus22\oplus23\oplus29\oplus30\oplus31\oplus18\oplus19)=11011\oplus$
$10110\oplus11000\oplus10111\oplus11101\oplus11110\oplus11111\oplus10010\oplus10011=11111$

–
$\grave{S}_{r3}=X-OR(26\oplus24\oplus25\oplus23\oplus17\oplus29\oplus31\oplus18\oplus19\oplus20)=$
$11010\oplus11000\oplus11001\oplus10111\oplus10001\oplus11101\oplus11111\oplus10010\oplus10011\oplus10100=10100$

–$\grave{S}_{r4}=X-OR(28\oplus29\oplus21\oplus22\oplus23\oplus26\oplus18\oplus27)=$
1100
$\oplus11101\oplus10101\oplus10110\oplus10111\oplus11010\oplus10010\oplus11011=00110.$

Experimental Results

The plat form of our research is Celeron 450MHz processor, 256 MBR am, windows XP professional operating system, Matlab programming, and adobephotoshop7. 0 as a graphical tool. In our experiments, we use four 256×256 binary images. These images are *Mandrill, geometrical shape, Mickey and English text*. We compare the performance of our algorithm with the CPT algorithm from two points so f views:

a. Quality: quality of the images is determined by row eye judgment, after hiding data in the host binary image.
b. Capacity: the amount of data that the host $m\times n$ block size can hide.

In Fig., 11, we show the effect of hiding data in the binary

image *Mandrill* using the CPT and the propose algorithm. In the two algorithms, precaution is taken not to hide in the black or white image blocks. Fig.11 (a) presents the original host image. In the figures 11 (b), 11 (c), and 11 (d) we use the CPT algorithm to hide 650 bytes with block size 8×8, 237 bytes with block size 16×16, and 128 bytes with block size 32×32 respectively. Similarly using the proposed algorithm in figures 11 (e), 11 (f), and 11 (g), we hide 653 bytes with block size 8×8, 266 bytes with block size 16×16, and 141 bytes with block size32×32.

From Fig. 11, it is easy to see that the proposed algorithm hides more data than the CPT algorithm, while keeping the same quality level for the stego image. To ensure these results, we repeated the experiments using the binary images *geo metrical shape, Mickey, and English text*. In tables 1,2, and 3, we list he capacity that we could hide using the proposed and the CPT algorithms for different block sizes. Based on obtained results, we can say that the proposed algorithm hides more data than the CPT algorithm while keeping the same quality level for stego image.

Table1: The results after taking block size of size 8×8

	Proposed Capacity	**CPT Capacity**
Geometrical shape	196byte	168byte
Mickey	477byte	409byte
English text	596byte	512byte
Mandrill	653byte	650byte

Table 2:The results after taking block size of size 16×16

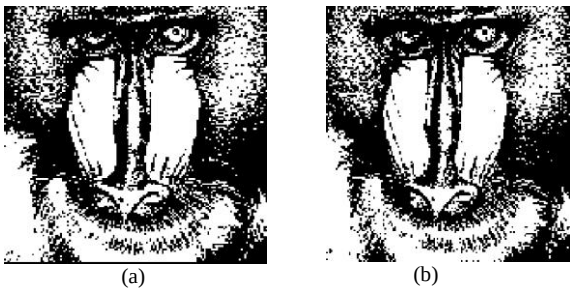
	Proposed Capacity	**CPT Capacity**
Geometrical shape	72byte	64byte
Mickey	166byte	148byte
English text	159byte	142byte
Mandrill	266byte	237byte

Table3:theresultsaftertakingblocksizeofsize32×32

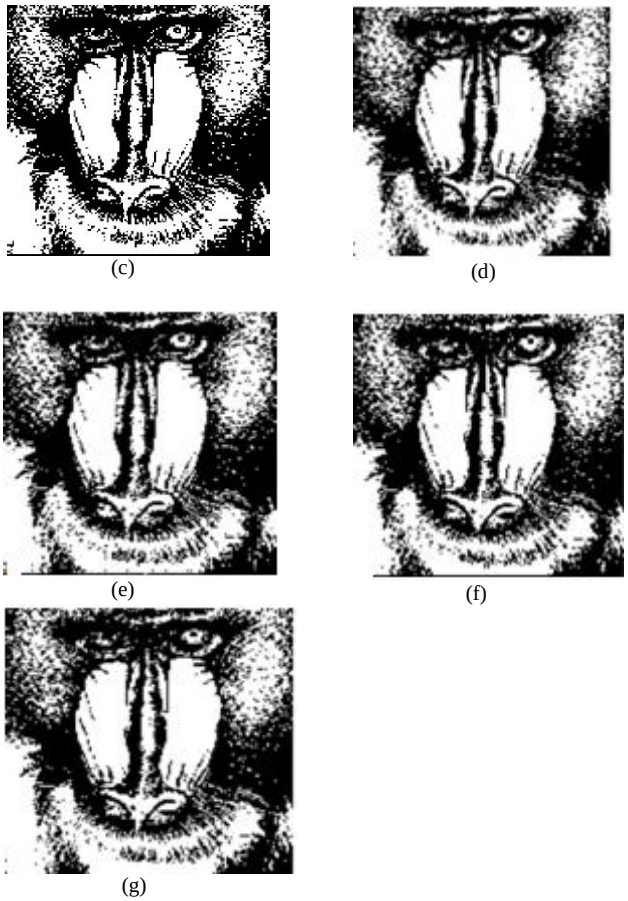
	Proposed Capacity	**CPT Capacity**
Geometrical shape	97byte	88byte
Mickey	144byte	131byte
English text	189byte	172byte
Mandrill	141byte	128byte

CONCLUSIONS

We have presented a new steganography scheme for hiding data in a host binary image. A secret key, and a new form of dynamic weight matrix were introduced to protect the hidden data, and to increase its capacity. The weight matrix changes its position at every block, as a result, the security



(a) (b)



(c) (d) (e) (f) (g)

Figure. (11): hiding effect

Figure 11: hiding effect on Mandrill image (a) the original host image, (b) hiding 650 bytes by CPT with block size 8 x 8 , (c) hiding 237 bytes by CPT with block size 16 x 16, (d) hiding 128 bytes by CPT with block by CPT with block size 32 x 32 , (e) hiding 653 bytes by the proposed hiding scheme with block size 8 x 8, (f) hiding 266 bytes by the proposed hiding scheme 16 x 16, (g) hiding 141 bytes by the proposed hiding scheme with block size 32x32Of the proposed hiding scheme is increased. The experimental results show that the proposed scheme can hide more data and have higher security than the CPT scheme while keeping the same quality level of the stego image. Future research could involve encrypting the data before hiding.

REFERENCES

[1]. W. Stallings Cryptography and network security (Prentice Hall, New Jersey, 2003).

[2]. A. MS Public key cryptography, applications algorithms and mathematical explanations, (Tata Elxsi, India, 2007)

[3]. D. Kohn. The codebreakers : the story of secret of writing (Scrib, New York, 1996).

[4]. W. Stallings. Cryptography and network security (Prentice Hall, New Jersey, 1999).

[5]. N. Ferguson, B. Schneier Practical cryptography (John Wiley, 2003).

[6]. F.Petitcolas, R.Anderson, and M. Kuhn, Information hiding – A survey, Proc. IEEE, 87(7), 1999, 1062-1078

[7]. R.J.AndersonandF.A.P.Petitcolas,Onthelimitsofsteganography,IEEEJ.onSelectedAreasinCommunications,16(4),1998,474-481

[8]. E. Franzetal, Computer-based steganography, in information

hiding, Lecture Notes in Com- puterScience,1174,1996,7-21

[9]. Y. Tseng; Y. Chen; H. Pan, A se cure data hiding scheme for binary images,IEEETrans. OnComm.,50(8),2002,1227-1231

[10]. J. Zhaoand E. Koch, Embedding robust labels into images for copyright protection, Proc. first Int. Conf. on Intellectual Property Rights for Information, Knowledge and New Techniques, Austria,Vienna,1995,242-251.

[11]. M.Y.Wuand J.H.Lee,A novel data embedding method for two-color facsimile images, Proc. Of International Symposium on Multimedia Information Processing, Taiwan,R.O.C,1998.

[12]. R.G.Van Schyndel, A.Z. Tirkel, and C.F. Osborne, A digital watermark, Proc. Of IEEEInt. Conf. on Image Processing, Austin, 1994,86-90.

[13]. R.Z.Wang,C.F.L in, and J.C.L in, Image hiding by LSB substitution and genetic algorithm, Proc. Of International Symposium on Multimedia Information Processing, Taiwan, R.O.C,1998.

[14]. Y.-Y.Chen,H.-K.Pan, and Y.-C.Tseng, A secure data hiding scheme for two-color images, Proc. IEEE Symp. Computers and Comm., Antibes, France, 2000,750–755.

[15]. M.Y.WuandJ.H.Lee.A novel data embedding methodfortwo-color fac simile images. In Int. Symposium on Multimedia Information Processing, Taipei,Taiwan,1998.

