International Journal of Informatics and Information Systems ISSN 2579-7069
Vol. 5, No. 1, January 2022, pp. 01-15

A Data Hiding Method Based on Partition Variable Block Size with Exclusive-or Operation on Binary Image

Chyuan-Huei Thomas Yang [1,*], Yu-Tzu Wang [2], Po-An Chen [3], and Shih-Syuan You [4]

Hsuan Chuang University, Taiwan
[1] chyang@hcu.edu.tw*; [2] md1024008@umail.hcu.edu.tw; [3] md1044001@umail.hcu.edu.tw, [4] bd1014002@umail.hcu.edu.tw
* corresponding author

(Received: August 1, 2021 Revised: October 20, 2021 Accepted: December 3, 2021, Available online: January 29, 2022)

Abstract

In this paper, we propose a high capacity data hiding method applying in binary images. Since a binary image has only two colors, black or white, it is hard to hide data imperceptible. The capacities and imperception are always in a trade-off problem. Before embedding we shuffle the secret data by a pseudo-random number generator to keep more secure. We divide the host image into several non-overlapping (2n+1) by (2n+1) sub-blocks in an M by N host image as many as possible, where n can equal 1, 2, 3 , …, or min(M,N). Then we partition each sub-block into four overlapping (n+1) by (n+1) sub-blocks. We skip the all blacks or all whites in each (2n+1) by (2n+1) sub-blocks. We consider all four (n+1) by (n+1) sub-blocks to check the XOR between the non overlapping parts and center pixel of the (2n+1) by (2n+1) sub-block, it embed n 2 bits in each (n+1) by (n+1) sub-block, totally are 4*n 2 . The entire host image can be embedded 4×n 2×M/(2n+1)×N/(2n+1) bits. The extraction way is simply to test the XOR between center pixel with their non-overlapping part of each sub-block. All embedding bits are collected and shuffled back to the original order. The adaptive means the partitioning sub-block may affect the capacities and imperception that we want to select. The experimental results show that the method provides the large embedding capacity and keeps imperceptible and reveal the host image lossless.

Keywords: Data hiding; Adaptive; Steganography; XOR; Imperceptible

1. Introduction

The digital generation has arrived for many years. The modern information technology can convert any media to digital data, and to spread information through the Internet. The Internet becomes very important tools to receive information through the Internet, and share at the same time in living life. This caused the problems concerning with privacy, rights of authentic owners, tampering of data, and data theft. To protect the transmission of legal information on the Internet becomes an important topic. Data hiding technology is to hide secret messages in host media, and make sure that will not be significantly changed after the hiding. It can avoid the illegal people steals the secret data, and accomplish concealing secret data.

Data hiding is a very effective method used to transmit secret messages, but with the growth of computers, the Internet, and more miscellaneous communications methods. Data hiding technology is developed in advanced and various. In real life, people can use the programs steal secrets from the company's archives through e-mail, Web space or portable storage device delivery. Those staff within a company can easily use these methods to sell competitors or others. Despite the use of data hiding may have offended so much, but if applied to legitimate uses of data hiding, it can protect knowledgeable property, isolate secrets from stealing, or concealment for valuable information to prevent its obliteration, theft or unlawful viewing.

This paper is presented by five sections. Section one, we introduce the data hiding in binary image and mention those research background, motivation and research purposes. We review the references which discussed binary image Data hiding method and related our research. Section three, a new high capacity Data hiding method with adaptive

block size for binary images including the both rules of embedding and extraction methods are proposed. The fourth section shows the computer experimental results with various images. Finally, we give a summary.

2. The Proposed Method/Algorithm

Data hiding is to obscure the trustworthy information with a mechanism into digital multimedia products, such as images, graphics, text, audio, video, and so on. Data hiding does not change the original multimedia characters a lot, or make the incorrect perception. It cannot disclose the messages which hidden in it, so you can reach the purpose of confidential communication. Main goal for data hiding: large of amount information payload (Payload) to hide, stego-image quality is not too worse, and the secret data cannot be detected. A good data hiding technology of the core requirements are as follows: 1. Safety (Security): along with keys other than those that others can't read hidden secrets out of the image. 2. Invisible (Imperceptibility): host image into confidential information cannot gain too much distortion, namely stego-image quality must be above a certain level, so as to avoid the wrong questionable. 3. High load capacity (Payload): image quality requirements under the appearance of some maximum amount of hidden information in order to improve the utilization of information. However, information loading and stego-image quality problems is a trade-off question, when secret data are required when loading large, fake image quality will be greatly affected, so how to achieve a balance between the secret data loading and image quality, is the researchers' effort goal.

We classify the former research for data hiding method in binary or halftone image into two categories. Several solutions [2,7,9,10,12,16,18,19] are using the block operations, others [1, 4~6, 8, 11, 13~15, 17] are not. We also discuss those method applied the XOR operators [3, 18]. Due to the halftone image is created through a process called dithering, in which the density and pattern of black and white dots are varied to simulate different shades of gray. Thus the halftone images we may also look as a binary image.

Now we are going to discuss those authors who propose their work with blocks operations. Byun et al. [2] propose a data hiding method in binary images using optimized bit position to replace a secret bit. This method employs blocks, which are subdivided. The parity bit for a specified block decides whether to change or not, to embed a secret bit. By finding the best position to insert a secret bit for each divided block, the image quality of the resulting stego-image can be improved, while maintaining low computational complexity. The experimental results show that the proposed method has an improvement with respect to a previous work. Guo and Zhang [7] propose a data hiding scheme for binary images, including document images, halftone images, scanned figures, text and signatures with high capacity. In their scheme, the embedding efficiency and the placement of embedding changes are simultaneously. The upper bound of the number of bits that can be embedded in M×N image block is $n\log2((M\times N)/n + 1)$ by embedding at most n pixels. Huy and Kim [9] present an Improved Matrix Encoding (IME) scheme for hiding data into a binary image. Their scheme improved the CPT scheme. In the CPT scheme, each block F of q=m×n pixel of G is changed by at most two pixels for hiding data. CPT scheme's embedding rate is $r = \log2 (q + 1)$. The IME scheme is shown approximately as a $2r - 2$ embedding rate. They claim their method is better than Tseng-Pan's modified CPT scheme (MCPT) with $r - 1$ embedding rate. Jung et al. [10] propose a data hiding method for binary images using pixel-value weighting. The binary host image is partitioned into non-overlapping sub-blocks and judged the most suitable position to embed a secret bit for each sub-block.

They calculate a weighted value of a sub-block to select a pixel to be replaced. Lin [12] proposed encryption method employs sub-divided blocks by modified bit position to replace a secret bit. The subdivided block of the host binary image has three or more pixels. Each block hides a secret bit. Lin's method is fast calculation, simple, and with higher storage capacity. Wang et al. [16] propose a high capacity data hiding scheme for binary images based on block patterns in scanned images. The scheme proposes block patterns for a 2×2 block to enforce specific block-based relationships in order to embed a significant amount of data without causing noticeable artifacts. They introduce two kinds of matching pair (MP) methods, internal adjustment MP and external adjustment MP, designed to decrease the embedding changes. Shuffling is applied before embedding to reduce the distortion and improve the security. Our previous research, Yang et al., [18] proposed a reversible high capacity data hiding method applying in binary images. Since a binary image has only two colors, it is more difficult to hide data and imperceptible. It is a

trade-off between capacities and imperceptions in data hiding. We shuffle the secret data by a pseudo-random number generator (PRNG) before hiding to keep it more secure. The host image is divided into non overlapping four by four sub-blocks, which the secret data will be concealed. Then we partition each four by four sub-block into four overlapping three by three sub-blocks. We skip the all blacks or all whites in four by four sub-blocks.

We consider all four three by three sub-block to check the XOR between center and four corners. The extraction steps are simply to test the XOR between the four corner pixels and the centers of each three by three sub-block. All embedding bits are collected and shuffled back with PRNG to retrieve the original order. We use the same other host image to keep the record during embedding steps to reach the reversible. A method for embedding data in binary line drawing images is proposed by Zhang and Man [19]. Under the condition that the width of digital lines is greater than or equal to three, the method uses the constraints of line drawings, and represents the digital lines by using a set of 3×3 rational meshes. Those central pixels of the meshes are selected to be embedded and then to be extracted. Their proposed method can maintain higher quality of the host image after data hiding.

Other researchers propose those methods without block operation. Bandyopadhyay et al. [1] propose data hiding and extraction algorithms for handwritten signatures in binary images. According to their algorithm, size of the carrier image must be double (or more) the size of source image. Additional bytes or noise have to be filled into host image to make the required size. The header has to be updated by the new value, the sum of original size of file and amount of noise. It can be applied for ownership protection, copy control, annotation and authentication of digital media. The authors specifically focused on protection and authentication of handwritten signature. Cao and Kot [4] propose a novel data hiding method for authenticating binary images through establishing dense edge adaptive grids (EAG) for invariantly selecting good data carrying pixel locations (DCPL). They employ dynamic system structure with carefully designed local content adaptive processes (CAP) to iteratively trace new contour segments and to search for new DCPLs. By maintaining and updating a location status map, they redesign the fundamental content adaptive switch and a protection mechanism to preserve the local CAPs' contexts as well as their corresponding outcomes. Comparison shows that their method reaches better trade-off between large capacity and good perceptual quality. Feng et al. [5] propose a binary image data hiding scheme that aims to minimize the embedding distortion on the texture.

They extract the complement, rotation, and mirroring-invariant local texture patterns (crmiLTPs) from the binary image first. The weighted sum of crmiLTP changes when flipping one pixel is then employed to measure the flipping distortion corresponding to that pixel. The data hiding scheme generates the cover vector by dividing the scrambled image into superpixels. Then the syndrome-trellis code is applied to minimize the designed embedding distortion. The proposed data hiding scheme can achieve statistical security without degrading the image quality or the embedding capacity. A halftoning-based multilayer watermarking of low computational complexity is proposed by Gau et al. [6]. Another data-hiding technique is also applied to improve the security and embedding capacity. They used 256 reference tables to guarantee in halftone format. The watermarks are embedded by a set of optimized compressed tables for table lookup. When decoding, the least mean square metric is considered to increase the differences among those generated phenotypes of the embedding angles. They used the naïve Bayes classifier for classifying the associated angles to extract the embedded watermarks, then retrieving the additional hidden-layer watermarks. Binary image watermarking technology is the fundamental of other anti counterfeiting techniques. Hou et al. [8] proposed a novel watermarking technique for binary images. Firstly the embedding step in a binary image is split into multiple thumbnails using a perimeter expansion and sampling operation. Then embedding the information is by flipping pixels in the thumbnails. Finally, the watermarked binary image is produced by inverse sampling the thumbnails. The watermarking information is extracted by the difference between the number of black pixels in the thumbnail and the mean value.

Khan and Bhattacharya [11], their data hiding method aim to embed information secretly into a carrier data signal by revising the immaterial components for secret communication or copyright protection. The data-hiding operation caused to distortion the host image resulting in defective image resolution. These distortions are undesirable to appear, like military or medical images. The Edge-Adaptive grid (EAG) technique with data carrying pixel location (DCPL) gives the hidden location. Their method can attain reversibility, image recovery and data extractions. Lo et

al. [13] propose a novel scheme of data hiding for halftone image which is extended from Fu and Oscar's DHSPT method (data hiding smart pair toggling), it is called as DHADPT (data hiding by absolute difference pair toggling). They embed the binary data into the halftone image with reference to the original multi-toned image by evaluating the absolute difference between the neighboring gray level pixels. The location with minimum sum of the absolute differences of the eight neighboring locations will be the candidate to hide bit data. Their method has higher modified peak signal-to-noise ratio.

Naskar and Chakraborty [14] propose a lossless data hiding for halftone color images. Halftone images are widely used in the printing. The proposed method can embed information in the form of a binary bit sequence into halftone color images with negligible change in their perceptual quality, as well as recover the unmodified original host image after the hidden information has been extracted. The proposed method can be applied in reversible watermarking and steganography for authentication and content protection of colored halftone images. A novel blind method for hiding data in binary text images is presented by Tirandaz et al. [15]. In their method, the embedding process is limited to the edge pixels of all connected components. During embedding process, the secret data are located those positions to introduce minimal visual distortion. Unlike existing block-based methods, they embed the secret bits directly into outer boundary pixels of all connected components in the text image. Yu and Wang [17] propose a new method to embed data in binary images, including scanned text, figures, and cartoon images. The proposed method is based on the manipulating of chains, which are a representation of binary image, to hide data. The chain records the edge information of an image, and the modifications of chains are modifications of edge area. By employing the embeddable chains and using pseudo-random permutation, the method can embed a significant amount of data without causing perceptible artifacts. Next section we give our propose method.

3. Methodology

We are going to propose a block based with XOR operation by adaptive dividing block size high capacity data hiding method applying in binary images. Before embedding we shuffle the secret data by a pseudo-random number generator (PRNG) to keep more secure. We divide the host image into several non-overlapping (2n+1) by (2n+1) sub-blocks in an M by N host image as many as possible, where n can equal 2, 3 , …, or min(M,N). Such as a 10 by 10 image is divided into four 5 by 5 non overlapping blocks (n=2) in Fig. 3.1. Then we partition each sub-block into four overlapping (n+1) by (n+1) sub-blocks. Fig. 3.2 demonstrates four 3 by 3 overlapping sub-block. Fig. 3.2(a) is the upper left corner block, Fig. 3.2(b) is the upper right corner block, Fig. 3.2(c) is the bottom left corner block, and Fig. 3.2(d) is the bottom right corner block. We skip all blacks or all whites in each (2n+1) by (2n+1) sub-blocks, shown in Fig. 3.3. We consider all four (n+1) by (n+1) sub-blocks to check the XOR between each pixel of nonoverlapping parts and center pixel of the (2n+1) by (2n+1) sub-block, it embed n 2 bits in each (n+1) by (n+1) sub-block, totally are 4*n 2 embedded bits. The extraction way is simply to test the XOR between center pixel and their non-overlapping part of each sub-block. All embedding bits are collected and shuffled back by PRNG to the original order.

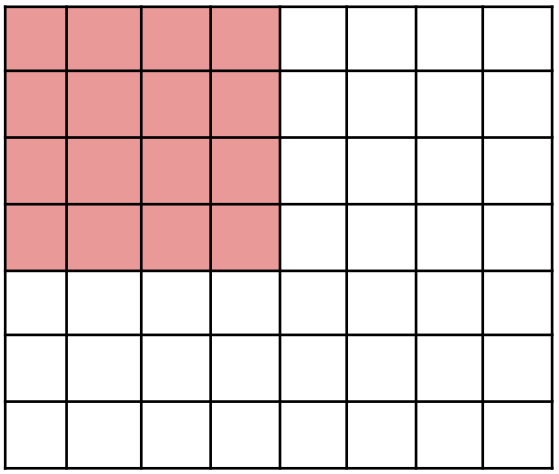
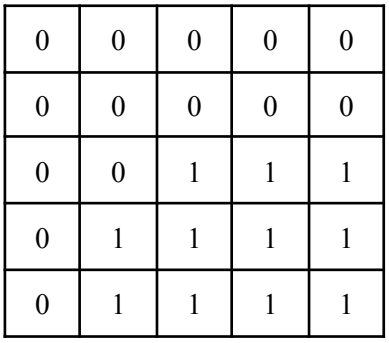
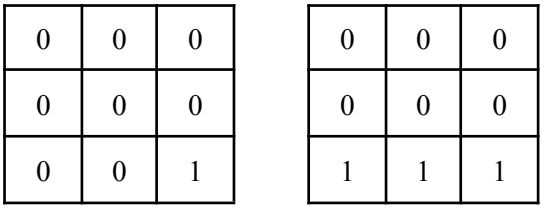
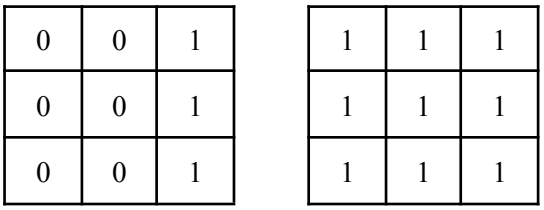


Figure. 1. The host image is divided into (2n+1)×(2n+1)block





(a) (b)



(c) (d)

Figure. 2. Each sub-block is divided into four overlapping (n+1)×(n+1) sub-blocks

3.1. An Example of Proposed Method

Assume that the host image size is $m \times n$, we may segment it into multiple non-overlapping 5×5 blocks B_i, that is $i = 1$ to$[m/5] \times [n/5]$. In each 5×5 block B_i we may divided it into four overlapping 3×3 sub-block B_{ij}, where $j = 1$ to 4, ai is the center pixels of 3×3 sub-block B_{ij}, B_j is the corner pixels of 3×3 sub-block B_{ij}. We apply a_i and b_j with XOR, if the result is 1, the embedded bit is 1, else the embedded bit is 0.

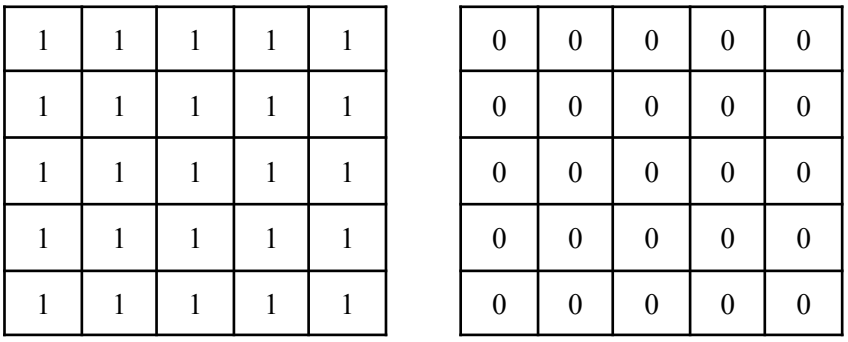


Figure. 3. All black and all white 5×5 block

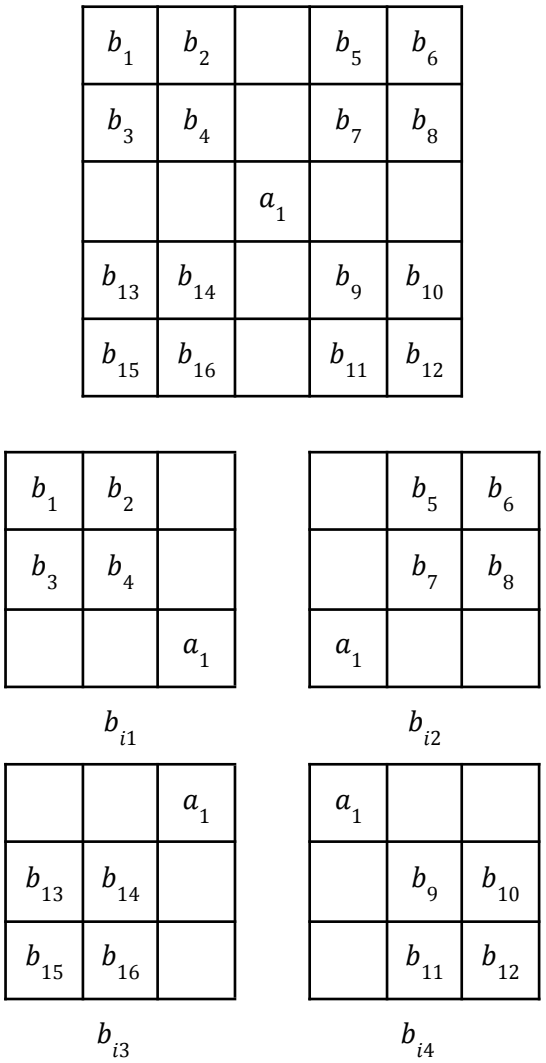


Figure. 4. The center pixel and its corner pixels of the four sub-blocks

We give an example to demonstrate shown in Figure 5. Assume b_i is the ith 5×5 block. This block is divided into four 3×3 blocks b_{i1}, b_{i2}, b_{i3}, b_{i4} (Figure 5(a)), Assume the embedding bit stream is [1 0 1 0] [1 1 1 0] [0 0 0 0] [1 0 0 1].

0	1	0	1	0
1	1	0	0	1
0	0	0	1	1
1	0	1	0	1
1	0	0	1	1

0	1	0
1	1	0
0	0	0

0	1	0
0	0	1
0	1	1

b_{i1}

0	0	0
1	0	1
1	0	0

b_{i2}

0	1	1
1	0	1
0	1	1

b_{i3} b_{i4}

(a) b_i is divided into four overlapping 3×3 sub-blocks b_{i1}, b_{i2}, b_{i3}, b_{i4}

0	1	0	→	1	0	0
1	1	0		0	1	0
0	0	0		0	0	0

(b) Embedding bits 1 0 1 0, $a_1 \oplus b_1 = 0$, $a_1 \oplus b_2 = 1$, $a_1 \oplus b_3 = 1$, $a_1 \oplus b_4 = 1$

0	1	0	→	0	1	1
0	0	1		0	0	1
0	1	1		0	1	1

(c) Embedding bits 1 1 1 0, $a_1 \oplus b_5 = 1$, $a_1 \oplus b_6 = 0$, $a_1 \oplus b_7 = 1$, $a_1 \oplus b_8 = 0$

0	1	1	→	0	1	1
1	0	1		1	0	0
0	1	1		0	0	0

(d) Embedding bits 0 0 0 0, $a_1 \oplus b_9 = 0$, $a_1 \oplus b_{10} = 1$, $a_1 \oplus b_{11} = 1$, $a_1 \oplus b_{12} = 1$

0	0	0	→	0	0	0
1	0	1		1	0	1
1	0	0		1	0	0

(e) Embedding bits 1 0 0 1, $a_1 \oplus b_{13} = 1$, $a_1 \oplus b_{14} = 0$, $a_1 \oplus b_{15} = 0$, $a_1 \oplus b_{16} = 1$

1	0	0	1	1
0	1	0	0	1
0	0	0	1	1
1	0	1	0	0
1	0	0	0	0

B_i

(f) The stego-image after embedded

Figure. 5. an example of embedding on a 5×5 block

3.2. Embedding Algorithm

For recording the length of secret data, we need an extra space to keep this length. We calculate the length of secret data, convert it into binary format, and put it before the scrambled bit scream of secret data. The data structure is shown in Figure 6. Sr is the converted bit stream of S, and scrambled by PRNG. $|S_r|$ is the length of S_r. $|L|$ is the bit numbers of L, that is the binary digits of $4 \times n^2 \times [M/(2n+1)] \times [N/(2n+1)]$. $M \times N$ is the size of host image. Thus the actual length of embedding data is $L + S_r$, that is $|L| + |S_r|$ binary digit. If we use a 512×512 host image, and n equals 2, the extra record space is no more than 18 bits. It is a few amount of bits, does not spent too much storage.

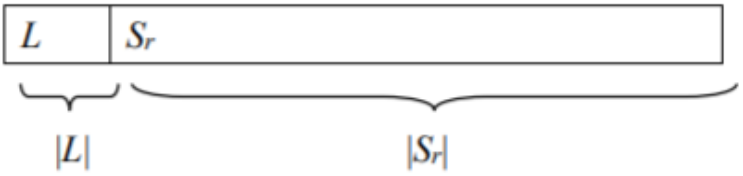


Figure. 6. The data structure of embedding data

Embedding Algorithm

a) Convert the secret data S into a bit stream, scramble it by PRNG to be S_r, the length of S_r is $|S_r|$

b) Convert $4 \times n^2 \times [M/(2n+1)] \times [N/(2n+1)]$ into a binary number L, $|L|$ is the number of bits of L. Put the L before S_r.

c) Divided the $m \times n$ host image C into non-overlapping (2n+1)×(2n+1) blocks B_i, where $i = 1$ to $[M/(2n+1)] \times [N/(2n+1)]$.

d) If there is no secret data left or embeddable block B_i, then STOP.

e) If B_i is an all blacks or all white, do nothing, go to next block B_i.

f) Divided B_i into four overlapping (n+1)×(n+1) sub-blocks B_{ij}, where $j = 1$ to 4.

g) If B_{ij} is all blacks or all whites, do nothing, go to next block B_{ij}.

h) If $a_i \oplus b_j$ equals to the current embedding bit in B_{ij}, do not adjust, that is the embedding bit is $a_i \oplus b_j$; if $a_i \oplus b_j$ is not equal to the current embedding bit in B_{ij}, we flip the b_j pixel from 0 to 1 or from 1 to 0.

i) Check the current (2n+1)×(2n+1) sub-block after embedding, if this sub-block is all blacks or all white, flip the center pixel a_i. Go back to Step 8.

j) Go to Step 4.

3.3. Extraction Algorithm

Basically the extraction procedure is similar to the embedding processes. Firstly we divide the stego-image into (2n+1)×(2n+1) non-overlapping blocks, skip all blacks or all whites. If (2n+1)×(2n+1) block is not all blacks or all whites then it is divided into four n×n sub-blocks. If the (n+1)×(n+1) sub-block is not all blacks or all whites then it is applied center pixel and three corner pixels with XOR to retrieve the secret data. After collecting all secret bits, shuffling back by PRNG, we may obtain the original secret data.

We give an example shown in Figure 7. Assume B_i is the ith 5×5 block, the procedure to extract as follows. From the first sub-block B_{i1}, we have $a_i \oplus b_1 = 1$, $a_i \oplus b_2 = 0$, $a_i \oplus b_3 = 1$, and $a_i \oplus b_4 = 0$. The second sub-block B_{i2}, we have $a_i \oplus b_5 = 0$, $a_i \oplus b_6 = 0$, $a_i \oplus b_7 = 0$, and $a_i \oplus b_8 = 1$. The third sub-block B_{i3}, we have $a_i \oplus b_9 = 1$, $a_i \oplus b_{10} = 0$, $a_i \oplus b_{11} = 1$, $a_i \oplus b_{12} =$

0. The fourth sub-block B_{i4}, we have $a_i \oplus b_{13}=0$, $a_i \oplus b_{14}=1$, $a_i \oplus b_{15}=0$, and $a_i \oplus b_{16}=1$. Thus we extract this 5×5 block to get the binary bit stream 1010 0001 1010 0101.

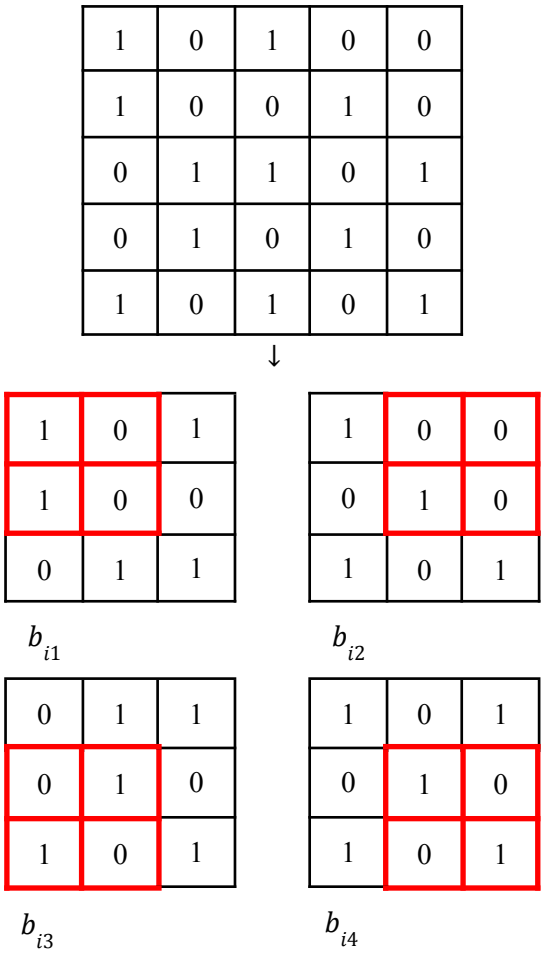


Figure. 7. The result of extraction example

Extraction Algorithm

a) Convert $4 \times n^2 \times [M/(2n+1)] \times [N/(2n+1)]$ into binary number L, $|L|$ is the length of L.
b) Divide m×n stego-image C into non-overlapping n×n blocks B_i, where $i=1$ to $[M/(2n+1)] \times [N/(2n+1)]$.
c) Skip the processing of B_i, if it is all blacks or all whites.
d) Divide B_i into four overlapping (n+1)×(n+1) sub-blocks B_{ij}, where $j=1$ to 4.
e) Apply $a_i \oplus b_j$ to extract the first $|L|$ bits, convert into decimal numbers. It is the length of embedded data $|S_r|$.
f) Apply $a_i \oplus b_j$ to find the secret bit, concatenate to the bit stream S_r.
g) Apply the same PRNG in embedding procedure to shuffle back the order of S_r to restore the original bit stream S_r.
h) Convert the S_r into the original format of secret data S.

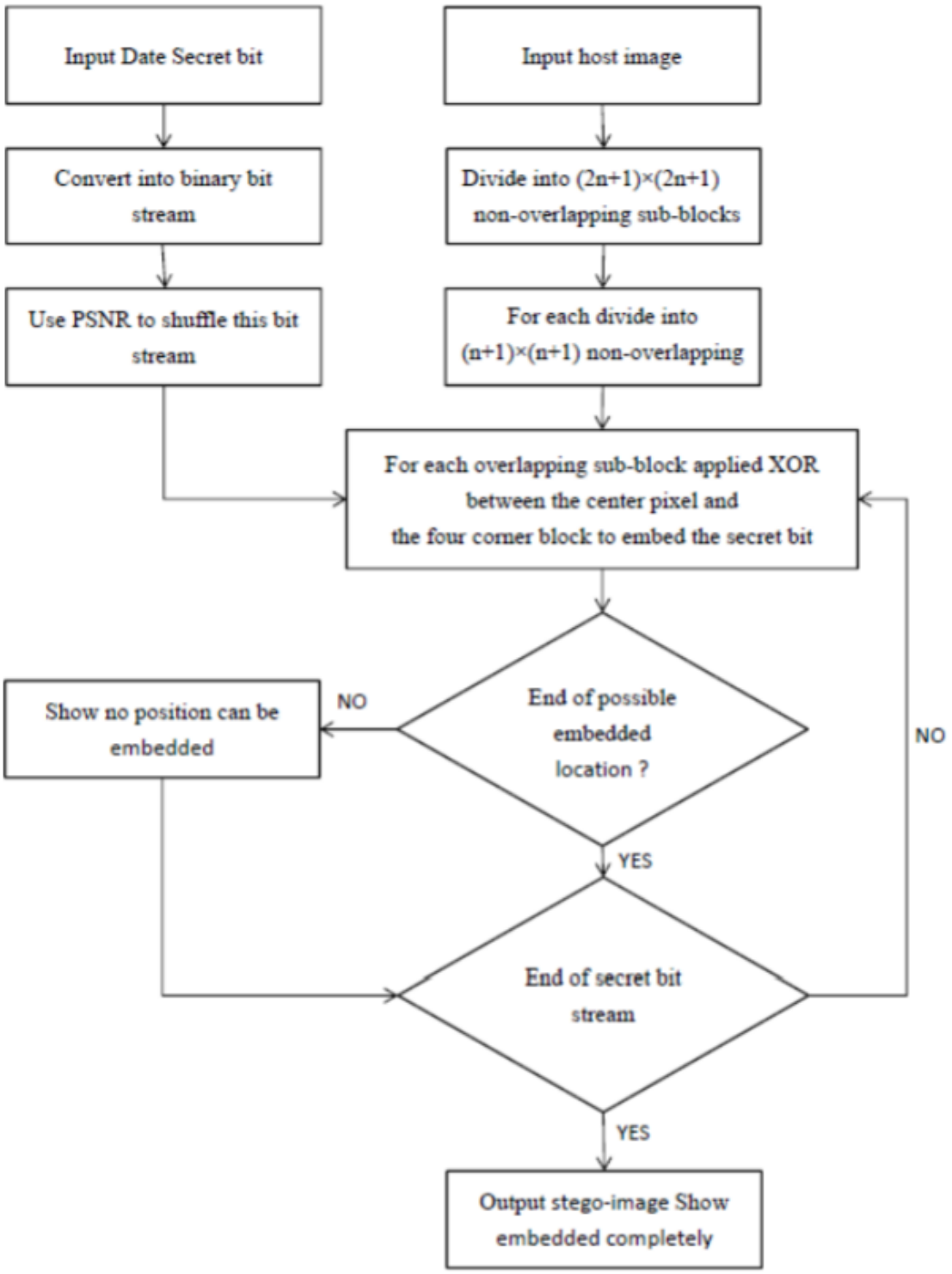


Figure. 8. The flowchart of embedding algorithm

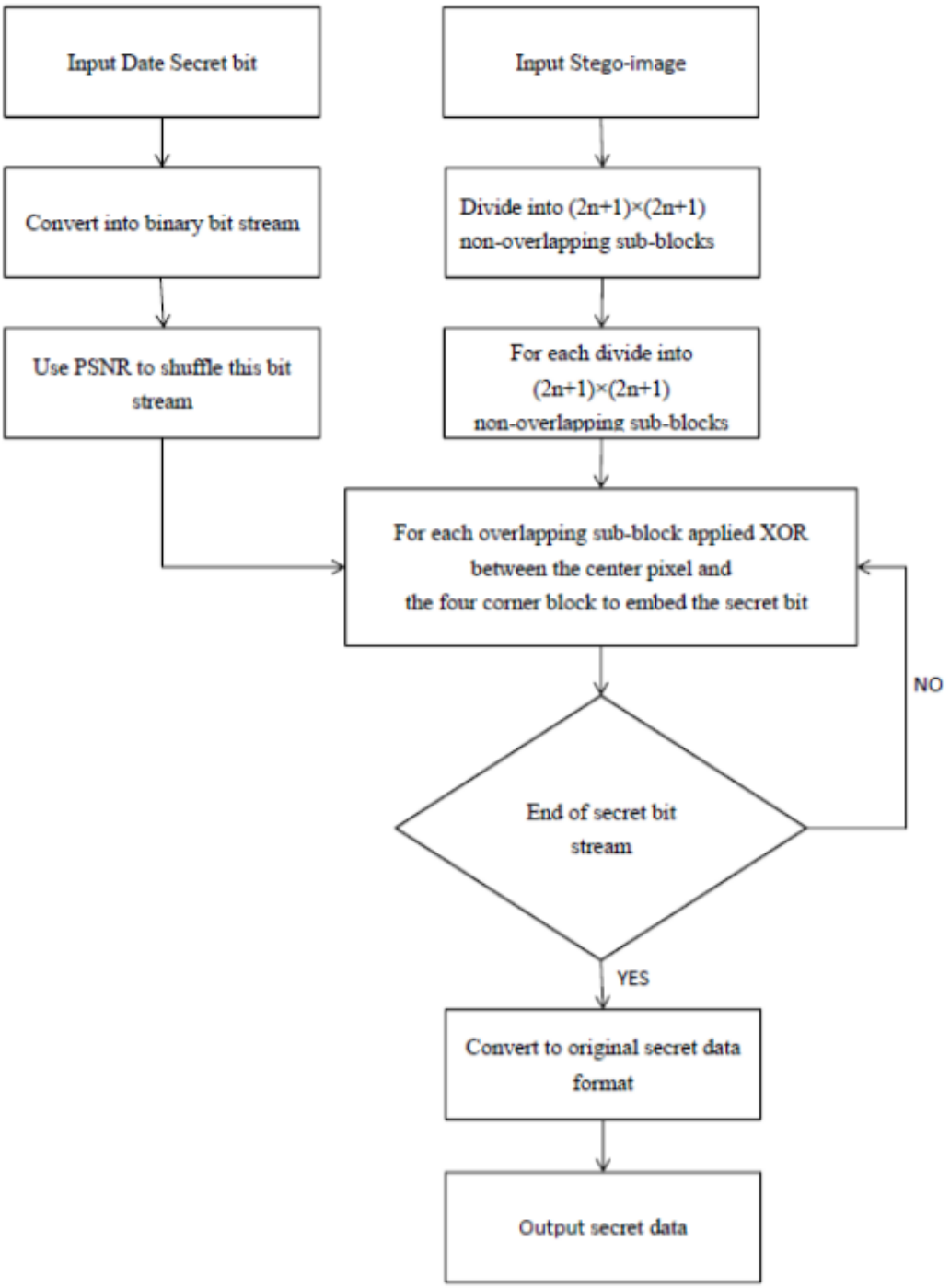


Figure. 9. The flowchart of extraction algorithm

4. Computer Experiments

In this section we demonstrate the experimental results. We use two logos of Hsuan Chuang University (Figure 10) to be the confidential (secret) data. The size of logo is 100×100. It is a binary image. We also use some popular images (Figure 11) to apply the computer experiments. These images many authors like to use them to test their methods. Also we use the Root Mean Square Error (RMSE) to examine the image quality. The equation of RMSE is

$$RMSE = \frac{\sqrt{\sum_{ij}\left(stego\ image_{ij} - cover\ image_{ij}\right)^2}}{m \times n} \quad (1)$$

i and *j* is the position in the image, *m x n* is the image size. Since binary images only need one bit to represent one pixel, we do not use the Peak Signal to Noise Ratio (PSNR) to be the image quality examination equation.

Figure. 10. Logo of Hsuan Chuang University (HCU)

Figure. 11. The cover images and their sizes (a) Cartoon:512×704 (b) Baboon :512×512 (c) Pepper:512×512 (d) Calligraphy:655×735 (e) Chinese newspaper:640×960 (f) English newspaper:570×500

Table 4.1 shows the RMSE of Embedded 100×100 HCU Logo in each cover image that we do computer experiment for our proposed method. Due to the size of logo is 100×100, thus the embedded bits should be 10000 bits, the extra 18 to 20 bits records the maximum possible length of embedded bits in each cover image. We divide sub-block size from 3 by 3 to the maximum possible block size. Fig. 4.3 shows the cover image of Cartoon is divided by different sub-block sizes, 3×3, 21×21 and 510×510 for each experiment.

Table. 1. RMSE of Embedded 100×100 HCU Logo in each cover image

Cover image	Image size	Embedded (Bits)	Average RMSE
Cartoon	512×704	10019	0.000151
Baboon	512×512	10018	0.000237
Pepper	512×512	10018	0.000235
Calligraphy	655×735	10019	0.000127
Chinese newspaper	640×960	10020	0.000097
English newspaper	570×500	10019	0.000206

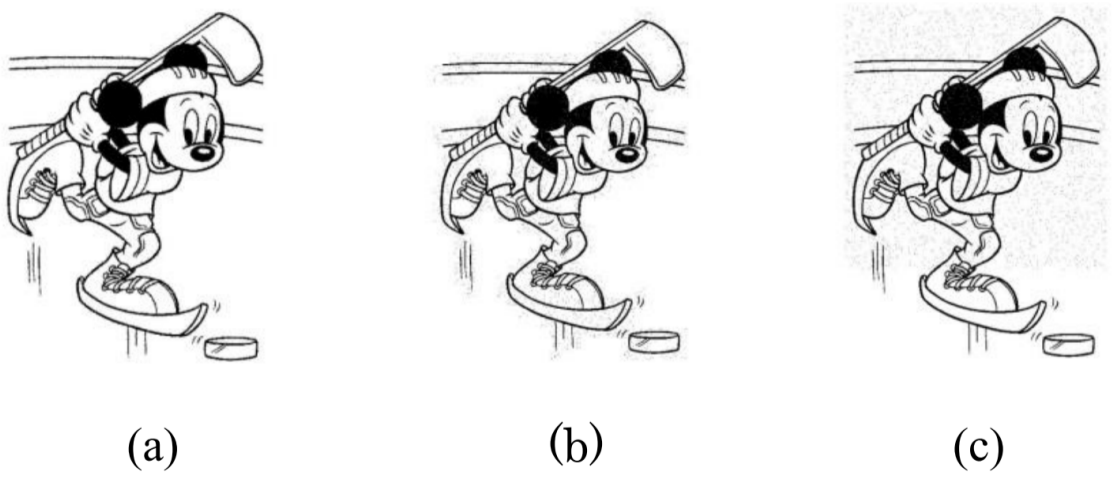


Figure. 12. (a) sub-block 3×3 size has the minimum RMSE 0.0002263 with embedded 6654 bits, (b) sub-block size 31×31 has the largest embedded 22414 bits with RMSE 0.0004153

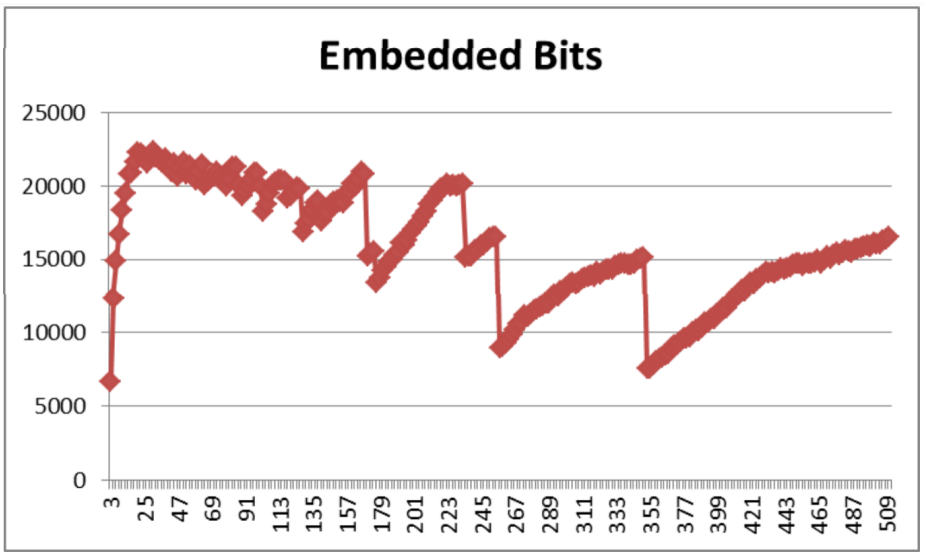


Figure. 13. The embedded bit for different sub-block sizes

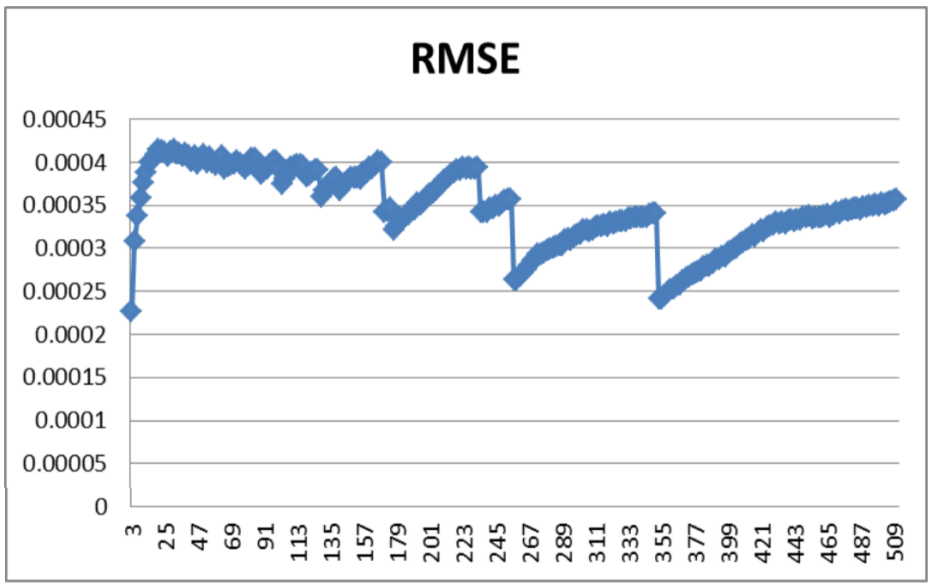


Figure. 13. The RMSE for each different sub-block sizes

Table. 2. The partitioned sub-block sizes of all cover images with the highest capacities and minimum RMSE

Cover Image	Image Size	Best RMSE			Maximum embedded		
		block size	bits	RMSE	block size	bits	RMSE
Cartoon	512x704	3 x 3	6654	0.000226	31 x 31	22414	0.000415
Baboon	512x512	257 x 257	14604	0.000461	73 x 73	52350	0.000873
Pepper	512x512	259 x 259	10718	0.000395	51 x 51	17912	0.000511
Calligraphy	655x735	369 x 369	13158	0.000238	53 x 53	43848	0.000435
Chinese newspaper	640x960	321 x 321	31946	0.000291	29 x 29	86786	0.000479
English newspaper	570x500	287 x 287	7580	0.000305	71 x 71	24140	0.000545

5. Conclusion

This paper proposed a data hiding method based on variable divided block size with exclusive-or operation on binary image. The trade-off problem is always in data hiding between high capacity and imperception. We divide the host image into several non-overlapping (2n+1) by (2n+1) sub-blocks in an M by N host image as many as possible, where n can equal 1, 2, 3 , …, or min(M,N). Then we partition each sub-block into four overlapping (n+1) by (n+1) sub-blocks. We skip the all blacks or all whites in each (2n+1) by (2n+1) sub-blocks. We consider all four (n+1) by (n+1) sub-blocks to check the XOR between the non overlapping parts and center pixel of the (2n+1) by (2n+1) sub-block, it embed n 2 bits in each (n+1) by (n+1) sub-block, totally are 4*n 2 . The entire host image can be embedded 4×n 2×M/(2n+1)×N/(2n+1) bits. For security, the proposed method uses pseudo-random number generators to first disrupt the order of confidential data, so that confidential data are not easily distinguishable to strengthen the security of confidential data. The partitioned sub-block size with the highest capacity of each cover image is different, so does the best imperceptions that we reach the adaptive meaning. We may consider in the future directions from the choice of the embedding starting point and the selection of pair points for mutually exclusive to minimize distortion of the images and produce the better stego-images.

References

[1] S.K. Bandyopadhyay, D. Bhattacharyya, D. Debnath and P. Das. (2008) “Bi-Color Nonlinear Data Embedding and Extraction of Handwritten Signature” Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on. IEEE, 2008.p1896 -1901.

[2] J. Y. Byun, K. H. Jung and K.Y. Yoo. (2008) “Improve Data Hiding Method by Block Parity for Binary Images” Computer Science and Software Engineering, 2008 International Conference on (Volume:3). IEEE, 2008.p 931 - 934.

[3] C. C. Chang, Y. H. Huang and W. C. Chang. (2015) “Reversible data hiding for VQ indices using XOR operator and SOC codes” Machine Learning and Cybernetics (ICMLC), 2015 International Conference on . IEEE, 2015.p 340 - 345.

[4] H. Cao. and A.C. Kot (2012) “EAG: Edge Adaptive Grid Data Hiding for Binary Image Authentication” Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific. IEEE, 2012.p1 –6.

[5] B. Feng, W. Lu and W. Sun.(2014) “Reversible Data Hiding Method Based on Exclusive-OR with Two Host Images” Trustworthy Systems and their Applications (TSA), International Conference on. IEEE, 2014.p 69 – 74.

[6] J.M. Guo, G.H. Lai,K.S. Wong and L.C. Chang(2015) “Progressive Halftone Watermarking Using Multilayer Table Lookup Strategy” Image Processing, IEEE Transactions on (Volume:24 , Issue: 7) .IEEE, 2015.p 1057-7149.

[7] M. Guo and H. Zhang. (2010) “High Capacity Data Hiding for Binary Image Authentication” Pattern Recognition (ICPR), 2010 20th International Conference on. IEEE, 2010.p 1441 - 1444.

[8] Q. Hou, D. Junping, Li Li, J. Lu And C.C. Chang.(2014) “Scanned binary image watermarking based on additive model and sampling” Springer Science+Business Media New York ,2014 Multimedia Tools and Applications, Multimedia Tools and Applications,2014. Volume 74, Issue 21 , pp 9407-9426.

[9] P. T. Huy1 and C. Kim. (2013) “Binary Image Data Hiding Using Matrix Encoding Technique in Sensors” International Journal of Distributed Sensor Networks Volume 2013, Article ID 340963, 7 pages.

[10] K.H. Jung, K.S. Ha and K.Y. Yoo. (2008) “Data Hiding in Binary Images by Pixel-valueWeighting” International Conference on Convergence and Hybrid Information Technology 2008, 2011 Seventh International Conference on. IEEE, 2008.p 262 - 265.

[11] S. Khan and A. Bhattacharya. (2015) “Secure Data Pixels for Binary Host Images Using Edge-Adaptive Grid Technique” Communication Technologies (GCCT), 2015 Global Conference on . IEEE, 2015.p 691 - 695.

[12] K. T. Lin. (2011) “Data Encrypting In A Binary Image Base On Modified Data Hiding Method” Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2011 Seventh International Conference on. IEEE, 2011.p 69 - 72.

[13] C.C. Lo ,C.M. Lee, B.Y. Liao and J.S. Pan (2008) “Halftone Image Data Hiding with Reference to Original Multitone Image” Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP '08 International Conference on. IEEE, 2008.p265 - 268.

[14] R. Naskar and R. S. Chakraborty. (2011) “Lossless Data Hiding for Halftone Images” Image Information Processing (ICIIP), 2011 International Conference on. IEEE, 2011.p 1 - 6.

[15] H. Tirandaz, R. Davarzani and, M. Monemizadeh and J. Haddadnia. (2009) “Invisible and High Capacity Data Hiding in Binary Text Images Based on Use of Edge Pixels” 2009 International Conference on Signal Processing Systems. IEEE, 2009.p 130 - 134.

[16] C.C. Wang, Y.F. Chang, C.C. Chang, J.K. Jan and C.C. Lin, (2014) “A high capacity data hiding scheme for binary images based on block patterns” The Journal of Systems and Software 93. ScienceDirect, 2014.p152 –162.

[17] X.Y. Yu and A. Wang. (2009) “Chain Coding Based Data Hiding in Binary Images” Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP '09. Fifth International Conference on. IEEE, 2009.p 933 - 936.

[18] C.-H.T. Yang , Y. T. Hsu , C.C. Wu And J.W. Chang.(2014) “Reversible Data Hiding Method Based on Exclusive-OR with Two Host Images” Trustworthy Systems and their Applications (TSA), International Conference on. IEEE, 2014.p 69 – 74.

[19] H.B. Zhang and L. Man (2008) “Data hiding in binary line drawing images” Wavelet Analysis and Pattern Recognition, 2008. ICWAPR '08. International Conference on (Volume:1). IEEE, 2008.p 134 - 140.