# Secure Authentication Watermarking for Binary Images using Pattern Matching

[1]Mr.M.Venkatesan, [2]Mrs. P.MeenakshiDevi, [3]Dr. K.Duraiswamy, [4]Dr.K.Thyagarajah

[1]Assistant Professor, K.S.Rangasamy College of Technology, TamilNadu, INDIA.
[2]Assistant Professor, K.S.Rangasamy College of Technology, TamilNadu, INDIA.
[3]Dean, K.S.Rangasamy College of Technology, TamilNadu, INDIA.
[4]Principal , PSNA College of Engineering and Technology, TamilNadu, INDIA.

**Summary**

In image authentication watermarking, hidden data is inserted into an image to detect any accidental or malicious image alteration. In the literature, quite a small number of cryptography based secure authentication methods are available for binary images. In a cryptography based authentication watermarking, a message authentication code (or digital signature) of the whole image is computed and the resulting code is inserted into the image itself. This paper proposes a new authentication watermarking method for binary images. The main idea is to use the prioritized sub-blocks by pattern matching scheme to embed the code. Shuffling is applied before embedding to equalize the uneven embedding capacity. It detects any alteration while maintaining good visual quality for all types of binary images. The security of the algorithm lies only on the secrecy of a secret or private keys used.

***Key Words:***
*Authentication Watermarking – Binary Images – Pattern Matching – Shuffling.*

## 1. Introduction

Data hiding represents a class of processes used to embed data, such as copyright information into various forms of media such as image, audio, or text with a minimum amount of perceivable degradation to the "host" signal; its goal is not to restrict or regulate access to the host signal, but rather to ensure that embedded data remains inviolate and recoverable.

A watermarking technique makes use of a data-hiding scheme to insert some information in the host image, in order to make an assertion about the image later. In this paper, data hiding scheme simply means the technique to embed a sequence of bits in a still image and to extract it afterwards.

Watermarking techniques can be classified as either "robust" or "fragile." Robust watermarks are useful for copyright and ownership assertion purposes. They cannot be easily removed and should resist common image-manipulation procedures. On the other hand, fragile watermarks (or authentication watermarks) are easily corrupted by any image processing procedure. However, watermarks for checking the image integrity and authenticity can be fragile because if the watermark is removed, the watermark detection algorithm will correctly report the corruption of the image.

In a cryptography based authentication watermarking [15,16], an authentication signature (AS) is computed from the whole image and inserted into the image itself. In cryptography, an AS is called message authentication code (MAC) using a secret-key cipher or digital signature (DS) using a public/private-key cipher. An AS contains information about the host image content that may be checked to verify its integrity. However, inserting the MAC/DS alters the image and consequently alters its MAC/DS, invalidating the watermarking [21]. To avoid this problem, for continuous-tone images, many authentication techniques compute the AS from the image clearing the least significant bits (LSBs) and insert the AS in LSBs. In other words, those bits where the watermark is to be inserted are not taken into account when computing MAC/DS. The same idea can be applied to binary images for computing MAC/DS.

A possible use of this technique is to send authenticated faxes and documents over networks and the Internet. In this case, the receiver of a document can verify its integrity for a given originator.

## 2. Data Hiding and Authentication Watermarking

In the literature, there are many authentication-watermarking techniques for continuous-tone images [1-3]. Also, there are many techniques for data hiding in binary and halftone images [4-6]. There are many papers on data hiding in binary images [7-13] but there is no authentication watermarking.

However, quite a small number of secure authentication watermarking techniques are available for

binary and halftone images [14-22]. Only some techniques used the cryptography-based secure authentication watermarking. In [15], it was given for dispersed-dot halftone images but the visual quality for a binary image is poor. In [19], the ratio of black versus white pixels is used. Although the algorithm aims at robustly hiding information in binary image, it is not secure enough to be directly applied for authentication or other fragile use.

In [22], sometimes PWLC does not correctly extract the hidden data, and fails to recover perfectly the original cover image. In summary, these previously proposed approaches either cannot be easily extended to other binary images, or can only embed a small amount of data.

In secure authentication watermarking using some data hiding technique for binary image, one must compute a hashing function of the binary image *F*, obtaining the hash value $H = H(F)$. After encryption, it becomes MAC/DS. This MAC/DS must be inserted into F itself, obtaining the marked image F′.

In this paper, a Secure Authentication Watermarking Technique for binary images using Pattern Matching scheme is proposed (SAWT-PM). The hidden data can be extracted without using the original unmarked image. The approach can be used to verify whether a binary document has been tampered with or not and also authenticates the originator.

The original image is represented as F. It is partitioned into m × n sub blocks (say 3 × 3). In each sub block only the middle pixel is used to hide the information. Each 3 × 3 sub-block is checked against predefined patterns. If a sub-block matches with any of the valid patterns, it is ready to hide the information. It is referred as R*eadyBlock,* and the middle pixel of it can be used to hide the information.

But, the idea of computing an AS of the whole binary image and inserting it into the same image fails. Since the insertion will modify the image fingerprint, it can not be verified at the receiver side. A modified idea to insert the AS without modifying the image fingerprint is to divide the image into two regions: The first region is small in size, called AS Region (ASR) where AS will be inserted and the second region is the original image excluding ASR, called Non-AS Region (NASR) from where AS will be computed. Using this technique, the image authenticity can be verified at the receiver side [12, 15, 16].

## 3. The SAWT-PM

In SAWT-PM, only a few pixels are modified and the positions of sub-blocks containing those pixels are known both in the insertion and extraction phases. SAWT-PM flips only low-visibility pixels to hide the information and consequently the watermarked image has excellent visual quality and do not have salt-and-pepper noise.

### 3.1 Finding *ReadyBlocks*

The original image is represented as F. It is partitioned into m × n sub-blocks $F_i$. In each sub-block only the middle pixel is used to hide the information. But, not all the sub-blocks are used for hiding the information. Among the 512, 3 × 3 sub-blocks several blocks were rejected due to various reasons like, high visibility, nonreversible at receiver side etc. Only 120 patterns are considered to be valid for data hiding.

The Figure 1 depicts the valid patterns. The hatched middle pixel may either be black or white. The change of middle pixel is less noticeable. Mirrors, transposes and reverses of the patterns are also used for hiding the information.

Each 3 × 3 sub-block is checked against various constraints to identify that whether it matches with any of the valid 120 patterns or not. If a sub-block matches with any of the valid patterns, it is a R*eadyBlock.*

The patterns are selected based on the following constraints:

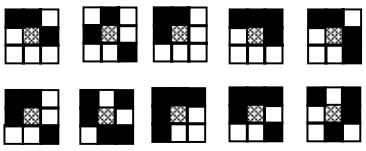


**Fig 1:** The 3 × 3 patterns used for hiding the information. The hatched middle pixel may either be black or white. The change of middle pixel is less noticeable. Mirrors, transposes and reverses of the above patterns are also used for hiding the information. The remaining patterns are not used.

(a) The number of white pixels in a sub-block should be greater than 2 and less than 6 excluding the middle pixel. That is, 3, 4, 5 or 6 white pixels and 6, 5, 4 or 3 black pixels respectively.

***I1:*** *X1 < SUM ($F_i$) < X2,* i =1 to n; X1=2; X2=6

where,

$F_i$ – a sub-block,
n – total number of sub-blocks
SUM – a function that counts the number of ones (white pixels) in the sub- block.
X1 and X2 – minimum and maximum number of ones in the sub-block respectively.

(b) If there are only 3 white/black pixels, they should not be in same row/column.

If a sub-block satisfies the above constraints, it will become R*eadyBlock,* and ready to hide the information.

**Authentication Signature**

Let k be the length of the adopted AS. To insert k bits of AS, it needs k, m × n *ReadyBlocks* in the image. The image is divided into two regions: The first region is small in size, called AS Region (ASR) where AS will be inserted and the second region is the original image excluding ASR, called Non-AS Region (NASR) from where AS will be computed. Using this technique, the image authenticity can be verified at the receiver side.

Once the *ReadyBlocks* are found, it is clearly understood that the middle pixels of all the *ReadyBlocks* forms the ASR. So, before calculating the hash value of the image, the ASR is made to be darkened. The new image is referred as (ZASR –Zeros inserted at AS Region).The ZASR is actually the whole image with zeros in ASR. Now, the hash value is calculated for ZASR, encrypted using the secret- or public key $k_s$, obtaining the MAC/DS and is inserted in ASR.

At the receiving side, the receiver uses the same technique to identify the R*eadyBlocks*. ASR is separated, encrypted AS inserted in that are retrieved and decrypted using the secret- or public key $k_s$, obtaining the AS. Then in the received original image, middle pixels of ASR are made to be darkened. It is referred as ZASR*. Compute H (ZASR*) and if H (ZASR*) =AS, then image integrity is verified. Otherwise, image has been modified or a wrong key was used.

### 3.3 Data Hiding Scheme

This technique ensures that for any pixel that is modified in the host image, its visibility is very less. Thus, the existence of secret information in the host image is difficult to detect.

The data hiding scheme is very simple. In each block, the position of the pixel to be used for hiding the information is fixed, that is, the middle pixel. Hence the complexity is not in identifying the position, but in identifying the sub-blocks that are ready to hide the information *(ReadyBlocks).*

Since the middle pixel is used to hide the information always, another level of security is introduced by shuffling of the *ReadyBlocks.* This provides another advantage also. It distributes the hidden information in all parts of the image. It is an efficient and effective tool to equalize uneven embedding capacity.

The *ReadyBlocks* are shuffled and permuted order is generated randomly. A secret key, called *Shuffling Key* shared by the sender and the receiver, is used as the seed for pseudo random number generator.

Let the original image F is partitioned into m × n sub-blocks $F_i$.

**Step 1:** If $F_i$ is completely black or blank, simply keep $F_i$ intact (not hidden with any information) and skip the following steps. Otherwise, perform the following:

**Step 2:** Identify the *ReadyBlocks* by comparing each sub-block $F_i$ with the predefined patterns specified in Section 3.1.

**Step 3:** Use the secret key to generate the permuted order of *ReadyBlocks.*

**Step 4:** If the middle bit of the *ReadyBlock* and the bit to be hidden are same, then no change is made to the sub-block $F_i$; otherwise, the middle bit is flipped to store the information.

### 3.4 Authentication Watermarking

The Figure 2 and Figure 3 depict the block diagram of embedding and extraction process in binary image authentication.

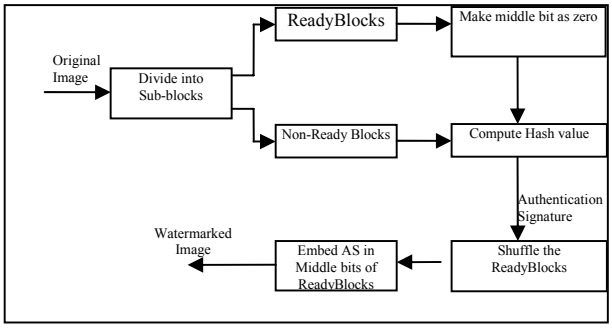



**Fig 2:** Block diagram of embedding process in binary image for authentication

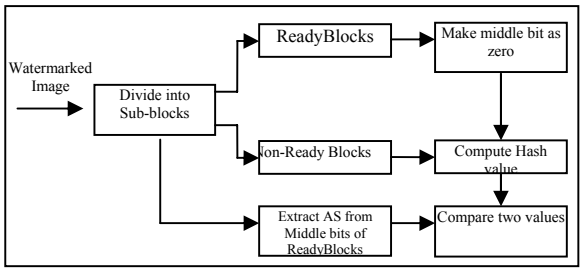



**Fig 3:** Block diagram of extraction process in binary image for authentication

The SAWT-PM insertion algorithm is:

1. Let *F* be a binary image to be watermarked and k be the length of AS.
2. Partition the image F into m × n size sub- blocks $F_i$. To insert k bits of AS, it needs k, m ×n blocks in the image.
3. Find the *ReadyBlocks* by comparing each sub-block $F_i$ with the predefined patterns specified.
4. The middle pixels of *ReadyBlocks* forms AS Region. Clear ASR, obtaining ZASR.
5. Compute the hash vale H=H (ZASR).
6. Encrypt H using the secret- or public key $k_s$, obtaining the digital signature S (MAC/DS).
7. Use the secret key to generate the permuted order of *ReadyBlocks*.
8. Insert S into ASR.

The SAWT-PM extraction algorithm is :

1. Let *F** be the watermarked image received.
2. Partition the image F into m × n size sub- blocks.
3. Find the *ReadyBlocks* by comparing each sub-block $F_i$ with the predefined patterns specified.
4. The middle pixels of *ReadyBlocks* forms AS Region. Clear ASR, obtaining ZASR*.
5. Compute the hash vale H*=H(ZASR*).
6. Use the secret key to generate the permuted order of *ReadyBlocks*.
7. Extract the watermark from the *ReadyBlocks* of F* and decrypt the result using the secret- or public key $k_s$, obtaining the digital signature S*
8. If S* and H* are equal the watermark is verified. Otherwise, the marked image *F** has been modified.

## 4. Experimental Results and Distortion Measure

In this section, experimental results are illustrated to demonstrate the validity of SAWT-PM data hiding scheme. Various binary images and document images, such as Mickey Mouse, English text are taken as the sample images in the experiment. The experimental results reveal that the SAWT-PM data hiding scheme exhibits very good performance.

### 4.1 Experimental results

Experimental results of some samples are given in Appendix – A. It shows the original image, watermarked image and modified pixels in the watermarked image of various samples taken. By observing the watermarked image, it is clear that the visual quality generated by CAPM data hiding scheme is very good.

#### 4.2 Distortion Measure

Two of the metrics used to compare the watermarked image and the original image are the Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR). The MSE is the cumulative squared error between the compressed and the original image, whereas PSNR is a measure of the peak error. It uses a standard mathematical

model to measure an objective difference between two images. The mathematical formulae for the two are,

$$\mathbf{MSE}= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \frac{(I(x,y) - I'(x,y))2}{M \times N}$$

$$\text{PSNR}= 20 \times \log 10 \frac{255}{sqrt(MSE)},$$

where I(x, y) is the original host image and I'(x, y) is the watermarked image and M,N are the dimensions of the images. A lower value for MSE means lesser error, and as seen from the inverse relation between the MSE and PSNR, this translates to a high value of PSNR. Logically, a higher value of PSNR is good because it means that the ratio of Signal to Noise is higher. Here, the 'signal' is the original image, and the 'noise' is the modified pixels in watermarked image. So, if a data embedding scheme having a lower MSE (and a high PSNR), it is recognized as a better one.

The PSNR and MSE obtained for the various categories of sample images are tabulated in Table 1. It reveals that watermarked images of good visual quality are obtained in the SAWT-PM data hiding scheme

**Table 1 - Experimental Result of Various Samples**

| Image | Image Size in bits | Total No. of m×n sub-blocks | No. of *Ready Blocks* | No.of Pixels Modified | PSNR in db | MSE |
|---|---|---|---|---|---|---|
| Mickey | 252×282 | 7896 | 330 | 89 | 29.0226 | 0.0013 |
| Text | 225×198 | 4950 | 400 | 83 | 27.2977 | 0.0019 |
| Monkey | 150×150 | 2500 | 274 | 80 | 24.4909 | 0.0036 |
| Flower | 143×143 | 2209 | 215 | 81 | 24.0219 | 0.0040 |
| Child | 123×93 | 1302 | 166 | 89 | 21.1946 | 0.0076 |

## 5. Conclusion

This paper has proposed a secure authentication watermarking for binary images using pattern matching (SAWT-PM). Shuffling is applied before embedding to equalize the uneven embedding capacity. The good performance is shown by experimental results. The proposed technique is suitable to watermark most binary images with good visual quality without causing a noticeable loss of quality. The applicable images include line-drawing images, cartoon images, maps etc. It can also be applied to provide basic proof of copyrights ownership and to electronically sign binary documents.

## 6. References


[1] P. S. L. M. Barreto, H. Y. Kim and V. Rijmen, "Toward a Secure Public-Key Block wise Fragile Authentication Watermarking," IEE Proc. Vision, Image and Signal Processing, vol. 149, no. 2, pp. 57-62, 2002.

[2] C. T. Li, D. C. Lou and T. H. Chen, "Image Authentication and Integrity Verification via Content-Based Watermarks and a Public Key Cryptosystem," IEEE Int. Conf. Image Processing, 2000, vol. 3, pp. 694-697.

[3] R. de Queiroz and P. Fleckenstein, "Object Modification for Data Embedding through Template Ranking," Xerox Invention Proposal, 1999.

[4] M. S. Fu and O. C. Au, "Data Hiding by Smart Pair Toggling for Halftone Images," IEEE Int. Conf. Acoustics, Speech and Signal Processing, vol. 4, pp. 2318-2321, 2000.

[5] M. S. Fu and O. C. Au, "Data Hiding Watermarking for Halftone Images," IEEE Trans. Image Processing, vol. 11, no. 4, pp. 477-484, 2002.

[6] S. C. Pei and J. M. Guo, "Hybrid Pixel-Based Data Hiding and Block-Based Watermarking for Error-Diffused Halftone Images," IEEE Trans. on Circuits and Systems for Video Technology, vol. 13, no. 8, pp. 867-884, 2003.

[7] Y.-C. Tseng, Y.-Y. Chen and H.-K. Pan, "A Secure Data Hiding Scheme for Binary Images," IEEE Trans. on Communications, Vol. 50, No. 8, Aug. 2002, pp. 1227-1231.

[8] G.Pass, Y.J.Wu and Z..h Wu, " A Novel Data Hiding Method for Two – color Images " , Lecture Notes in Computer Science Information and Communications Security, Springer-Verlag, NOV. 2001, PP. 261 – 270.

[9] M. Wu, E. Tang, and B. Liu, "Data hiding in digital binary image," IEEE Int. Conf. Multimedia & Expo (ICME'00), New York, 2000.

[10] Wen-Yuan Chen and Chen-Chung Liu, " Robust watermarking scheme for binary images using a slice-based large-cluster algorithm with a Hamming Code", Optical Engineering,Vol. 45, Iss.1, 2006.

[11] Jeanne chan, Tung-shan chen, Meng-wen cheng " A New Data Hiding Method in Binary Image" , Proc. IEEE Fifth International Symposium on Multimedia Software Engineering ( ISMSE ' 03 ). 2003.

[12] M.Venkatesan, P.Meenakshi Devi, K. Duraiswamy, K.Thiagarajah, "A New Data Hiding Scheme with Quality Control for Binary Images Using Block Parity", Information Assurance and Security, IAS 2007. Manchester, UK. pp. 468-471, 2007.

[13]Chang-Lung Tsai,Huei-Fen Chiang, Kuo-Chin Fan and Char-Dir Chung "Reversible data hiding and lossless reconstruction of binary images using pair-wise logical computation mechanism", Pattern

Recognition, vol. 38,issue 11, November 2005, Page 1993-2006.
[14] Min Wu, Member, IEEE, and Bede Liu, Fellow, IEEE, "Data Hiding in Binary Image for Authentication and Annotation", IEEE Transactions On Multimedia, Vol. 6, No. 4, August 2004.
[15] H. Y. Kim, A. Afif, "Secure Authentication Watermarking for Binary Images", in Proc. Sibgrapi-Brazilian Symp. On Computer Graphics and Image Processing,pp. 199-206,2003.
[16] Hae Yong Kim, Amir Afif, "A Secure Authentication Watermarking for Halftone and Binary Images", International Journal of Imaging Systems and Technology, Volume 14, Issue 4, Pages 147 – 152.
[17] M.Venkatesan, P.Meenakshi Devi, K. Duraiswamy, K.Thiagarajah, "Cryptography-based Secure Authentication Watermarking for Binary Images", CSI Communications, Vol.31, Issue No. 5, pp. 13-15. August 2007.
[18] M. U. Celik, G. Sharma, E. Saber and A. M. Tekalp, "Hierarchical Watermarking for Secure Image Authentication with Localization," IEEE Trans. Image Processing, vol. 11, No. 6, pp. 585-595, 2002.
[19] E. Koch and J. Zhao, "Embedding robust labels into images for copyright protection," in Proc. Int. Congr. Intellectual Property Rights forSpecialized Information, Knowledge & New Technologies, 1995.
[20] P. W. Wong, "A Public Key Watermark for Image Verification and Authentication," IEEE Int. Conf. Image Processing, vol. 1, pp. 455-459, 1998.
[21] Sergio Vicente D. Pamboukian, Hae Yong Kim, Reversible Data Hiding and Reversible Authentication Watermarking for Binary Images. SBSEG-Brazilian Symposium on Security of Information and Computing Systems, 2006, Santos.
[21] Hae Yong Kim and Ricardo Lopes de Queiroz, "Alteration-Locating Authentication Watermarking for Binary Images", International Workshop on Digital Watermarking, Springer Berlin / Heidelberg Volume 3304/2005, pp. 125-136,2005.
[22] Chang-Lung Tsai,Huei-Fen Chiang,Kuo-chin Fan,Char-Dir Chung,"Reversible data hiding and lossless reconstruction of binary images using pair-wise logical computation mechanism, Pattern Recognition,38(2005)1993-2006.

## Authors :

**Mr. M.Venkatesan** is working as Assistant Professor in Department of Computer Applications, K.S.Rangasamy College of Technology, TamilNadu, India. He received his B.Sc.(Electronics) in Sri Ramakrishna Mission Vidyalaya Arts and Science College, Coimbatore, in 1990 and M.C.A. degree in Bharathidhasan University, Trichy, in 1997. He is a research scholar in the Department of Computer Science and Engineering. His area of interest includes Watermarking, Information Security and Computer Networks.

**Mrs. P.Meenakshi Devi** is working as Assistant Professor in Department of Information Technology, K.S.Rangasamy College of Technology, TamilNadu, India. She received her B.E. degree in Kongu Engineering College, Perundurai, in 1993 and M.E. degree in Thiagarajar College of Engineering, Madurai, in 2003. She is a research scholar in the Department of Computer Science and Engineering. His area of interest includes Information Security, Watermarking, Cryptography and Computer Networks.

**Dr. K.Duraiswamy** received his B.E. degree in Electrical and Electronics Engineering from P.S.G. College of Technology, Coimbatore in 1965 and M.Sc.(Engg) from P.S.G. College of Technology, Coimbatore in 1968 and Ph.D. from Anna University in 1986. From 1965 to 1966 he was in Electricity Board. From 1968 to 1970 he was working in ACCET, Karaikudi. From 1970 to 1983, he was working in Government College of Engineering Salem. From 1983 to 1995, he was with Government College of Technology, Coimbatore as Professor. From 1995 to 2005 he was working as Principal at K.S.Rangasamy College of Technology, Tiruchengode and presently he is serving as Dean of KSRCT. He is interested in Digital Image Processing, Computer Architecture and Compiler Design. He received 7 years Long Service Gold Medal for NCC. He is a life

member in ISTE, Senoor member in IEEE and a member of CSI.

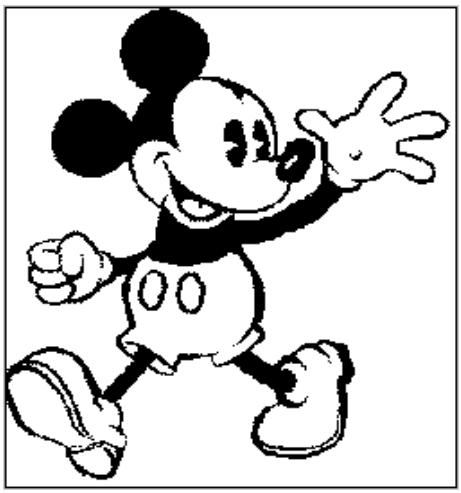
**Dr.K. Thyagarajah** received his B.E. degree in P.S.G. College of Technology, Coimbatore in 1976 and M.E. degree in College of Engineering in 1979. He received his Ph.D. form Indian Institute of Science, Bangalore in 1992. He worked in Regional Engineering College, Surathkal, K.S.Rangasamy College of Technology, Tiruchengode and presently he is working as a Principal in P.S.N.A. College of Engineering and Technology, Dindigul. His area of research includes, Power Electronics, Computer and Communications and Network Security. He has published over 20 research papers in National and International journals.

**[Appendix A Follows…]**

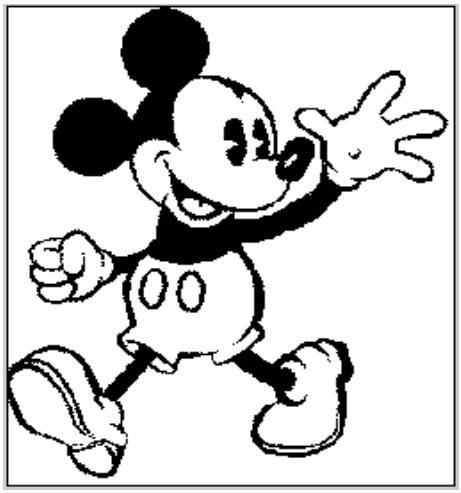
# Appendix – A

**Experimental Results of Some Sample Images**

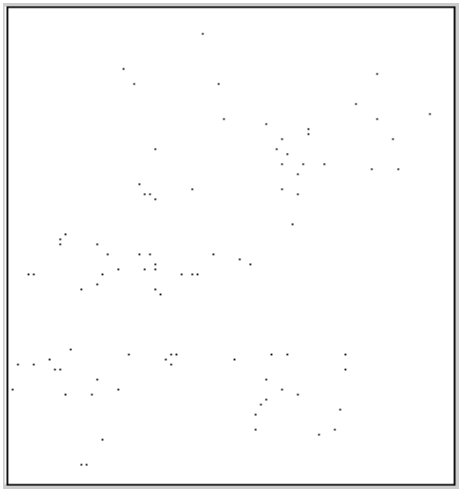
(a) original copy; (b) marked copy with 160-bit embedded in;
(c) difference between original and marked (shown in black)



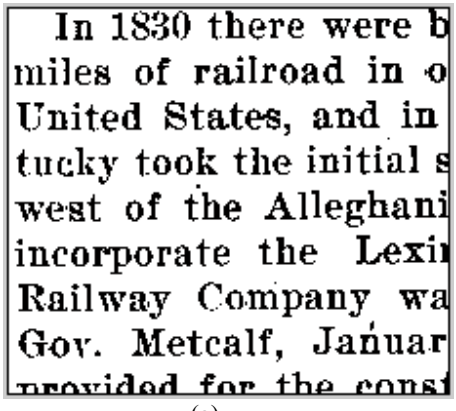
(a)



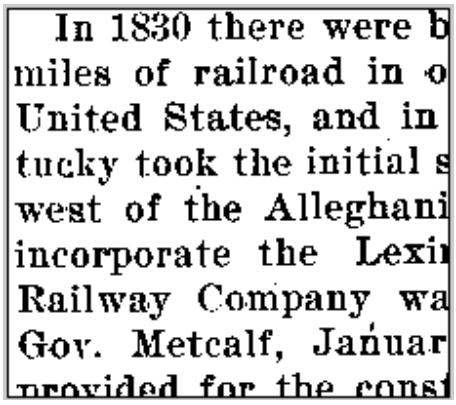
(b)



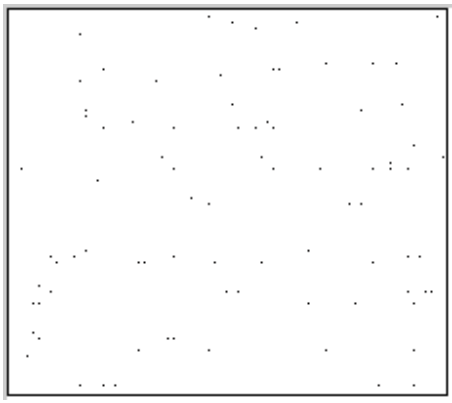
(c)



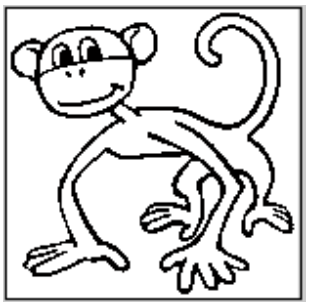

(a)


In 1830 there were b
miles of railroad in o
United States, and in
tucky took the initial s
west of the Alleghani
incorporate the Lexi
Railway Company wa
Gov. Metcalf, Januar
provided for the consi

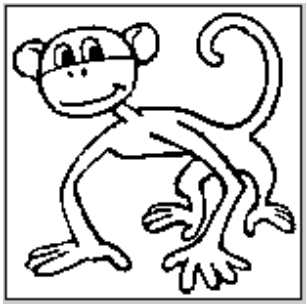
(b)



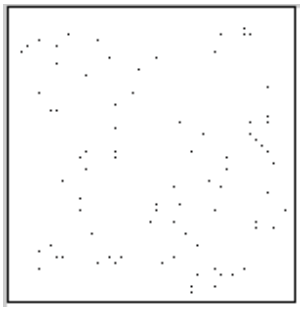
(c)



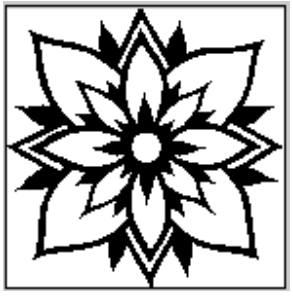
(a)



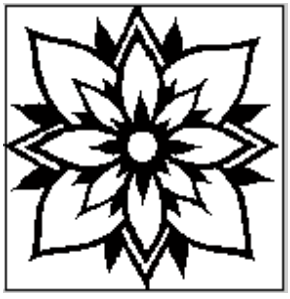
(b)



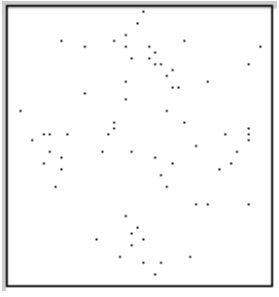
(c)



(a)



(b)



(c)

(a)

(b)

(c)