 ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

A Confidential Sharing and Repairng of Document Images

[1]T.Shailendra, [2]M.Madhava Rao
[1,2]Dept. of CSE, Sir CRR College of Engineering, Eluru, AP, India

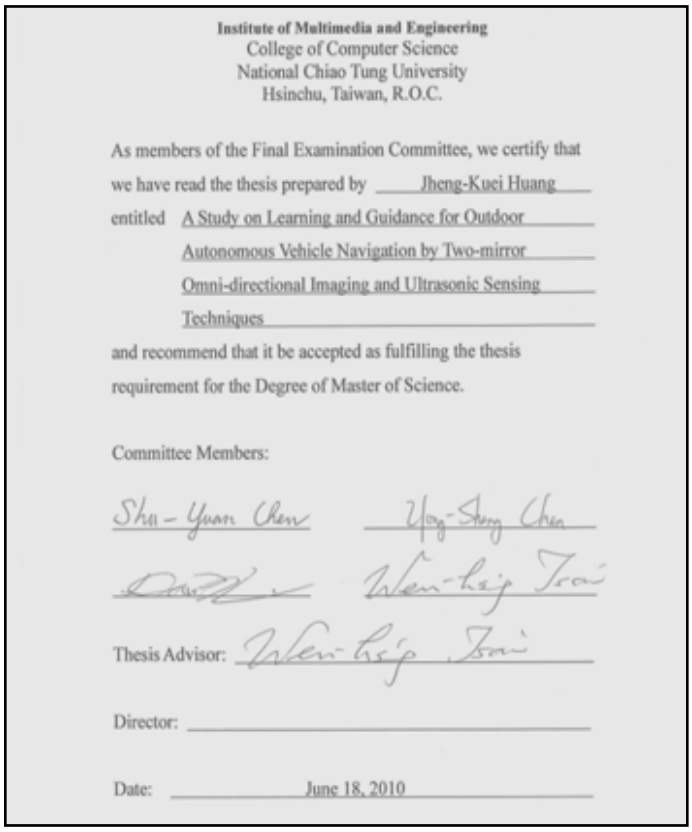
Abstract
An authentication signal is generated for each block of a grayscale document image, which, together with the binarized block content, is transformed into several shares using the Shamir secret sharing scheme. Many shares are generated and embedded into an alpha channel plane then the alpha channel is then combined with the original grayscale image to form a PNG image. During the embedding process, the computed share values are mapped into a range of alpha channel values near their maximum value of 255 to yield a transparent stego-image with a disguise effect. In the process of image authentication, an image block is marked as tampered if the authentication signal computed from the current block content does not match that extracted from the shares embedded in the alpha channel plane. Data repairing is then applied to each tampered block by a reverse Shamir scheme after collecting two shares from unmarked blocks.

Keywords
Data Repair, Grayscale Document Image, Image Authentication, Portable Network Graphics (PNG), Secret Sharing

I. Introduction

DIGITAL image is a form for preserving important infor- mation. However, with the fast advance of digital tech- nologies, it is easy to make visually imperceptible modifications to the contents of digital images. How to ensure the integrity and the authenticity of a digital image is thus a challenge. It is de- sirable to design effective methods to solve this kind of image authentication problem [1]–[3], particularly for images of doc- uments whose security must be protected. It is also hoped that, if part of a document image is verified to have been illicitly altered, the destroyed content can be repaired. Such image content authentication and self-repair capabilities are useful for the security protection of digital documents in many fields, such as important certificates, signed documents, scanned checks, circuit diagrams, art drawings, design drafts, last will and testaments, and so on.

Document images, which include texts, tables, line arts, etc., as main contents, are often digitized into grayscale images with two major gray values, one being of the background (including mainly blank spaces) and the other of the foreground (including mainly texts). It is noted that such images, although gray valued in nature, look like binary. For example, the two major gray values in the document image shown in Fig. 1 are 174 and 236, respectively. It seems that such binary-like grayscale document images may be thresholded into binary ones for later processing, but such a thresholding operation often destroys the smoothness of the boundaries of text characters, resulting in visually unpleasant stroke appearances with zigzag contours. Therefore, in practical applications, text documents are often digitized and kept as grayscale images for later visual inspection.


Institute of Multimedia and Engineering
College of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.

As members of the Final Examination Committee, we certify that we have read the thesis prepared by Jheng-Kuei Huang entitled A Study on Learning and Guidance for Outdoor Autonomous Vehicle Navigation by Two-mirror Omni-directional Imaging and Ultrasonic Sensing Techniques and recommend that it be accepted as fulfilling the thesis requirement for the Degree of Master of Science.

Committee Members:

Shi-Yuan Chen Yog-Sang Chen

Wen-hsiang Tsai

Thesis Advisor: Wen-hsiang Tsai

Director:

Date: June 18, 2010

Fig. 1: Binary-Like Grayscale Document Image With Two Major Gray Values

In general, the image authentication problem is difficult for a binary document image because of its simple binary nature that leads to perceptible changes after authentication signals are embedded in the image pixels. Such changes will arouse possible suspicions from attackers. A good solution to such binary image authentication should thus take into account not only the security issue of preventing image tampering but also the necessity of keeping the visual quality of the resulting image. In this paper, we propose an authentication method that deals with binary-like grayscale document images instead of pure binary ones and simultaneously solves the problems of image tampering detection and visual quality keeping.

Several methods for binary image authentication have been proposed in the past. Wu and Liu [4] manipulated the so-called flippable pixels to create specific relationships to embed data for authentication and annotation of binary images. Yang and Kot [5] proposed a two-layer binary image authentication method in which one layer is used for checking the image fidelity and the other for checking image integrity. In the method, a connec-tivity-preserving transition criterion for determining the flippability of a pixel is used for embedding the cryptographic signature and the block identifier. Later, Yang and Kot [6] proposed a pattern-based data hiding method for binary image authentica- tion in which three transition criteria are used to determine the flippabilities of pixels in each block, and the watermark is adap- tively embedded into embeddable blocks to deal with the uneven embeddability condition in the host image. In the method pro- posed in [7], a set of pseudorandom pixels in a binary or halftone image are chosen and

cleared, and authentication codes are ac- cordingly computed and inserted into selected random pixels. In Tzeng and Tsai's method [8], randomly generated authenti- cation codes are embedded into image blocks for use in image authentication, and a so-called code holder is used to reduce image distortion resulting from data embedding. Lee et al. [9] proposed a Hamming-code-based data embedding method that flips one pixel in each binary image block for embedding a wa- termark, yielding small distortions and low false negative rates. Lee et al. [10] improved the method later by using an edge line similarity measure to select flippable pixels for the purpose of reducing the distortion.

In this paper, a method for the authentication of document images with an additional self-repair capability for fixing tam- pered image data is proposed. The input cover image is as- sumed to be a binary-like grayscale image with two major gray values like the one shown in Fig. 1. After the proposed method is applied, the cover image is transformed into a stego-image in the Portable Network Graphics (PNG) format with an addi- tional alpha channel for transmission on networks or archiving in databases. The stego-image, when received or retrieved, may be verified by the proposed method for its authenticity. Integrity modifications of the stego-image can be detected by the method at the block level and repaired at the pixel level. In case the alpha channel is totally removed from the stego-image, the entire re- sulting image is regarded as inauthentic, meaning that the fi- delity check of the image fails. The proposed method is based on the so-called (k, n)-threshold secret sharing scheme proposed by Shamir [11] in which a secret message is transformed into n shares for keeping by participants, and when k of the n shares, not necessarily all of them, are collected, the secret message can be losslessly recovered. Such a secret sharing scheme is useful for reducing the risk of incidental partial data loss.

II Review of the Shamir Method For Secret Sharing

In the (k, n)-threshold secret sharing method proposed Shamir [11], secret d in the form of an integer is transformed into shares, which then are distributed to n participants for them to keep; and as long as k of the shares are collected, the original secret can be accordingly recovered, where $k \leq n$.

Algorithm 1: (k,n) -threshold secret sharing.
Input: secret d in the form of an integer, number n of participants, and threshold $k \leq$ n.
Output: n shares in the form of integers for the n participants to keep.
Step 1. Choose randomly a prime number p that is larger than d
Step2: Select k-1 integer values c_1, c_2.....c_{k-1} with in the range of 0 to p-1
Step3: Select n distinct real values x_1, x_2....x_n
Step4: Use the following (k-1) degree polynomial to compute n function values $F(x_i)$ called partial shares for i = 1,2,……..n, i.e.,

$$F(x_i) = (d+c_1x_i+c_2x_i^{2}+\ldots+c_{k-1}x_i^{k-1}) \bmod p \quad (1)$$

Step 5: Deliver the two tuple (xi, F(xi)) as a share to the i th participate where i =1,2, 3,…..n
Since there are k coefficients, namely, d and c_1 through c_{k-1} in (1) above, it is necessary to collect at least k shares from the n participants to form k equations of the form of (1) to solve these k coefficients in order to recover secret d

Algorithm 2: Secret recovery
Input: k shares collected from the n participants and the prime number p with both k and p being those used in Algorithm 1.
Output: Secret d hidden in the shares and coefficients ci used in (1) in Algorithm 1, where i = 1, 2 …….. k-1.
Steps
Step1: Use the k shares (x_1, $F(x_1)$), (x_2, $F(x_2)$… (x_k, $F(x_k)$) to set up

$$F(x_j) = (d+c_1x_j+c_2x_j^{2}+\ldots+c_{k-1}x_{j\,(k-1)} \bmod p \quad (2)$$

Where j = 1, 2,…..k
Step 2: Solve the equation by Lagrange's interpolation to obtain d as follows

$$d = (-1)^{k-1}((F(x_1)(x_2x_3\ldots x_k)/(x_1-x_2)(x_1-x_3)\ldots(x_2-x_k) + ((F(x_2)(x_1x_3\ldots x_k)/(x_2-x_1)(x_2-x_3)\ldots(x_2-x_k)\ldots + F(x_k)(x_1x_2\ldots x_{k-1})/(x_k-x_1)(x_k-x_2)\ldots(x_k-x_{k-1}))\bmod p$$

Step 3: Compute C_1 through C_{k-1} by expanding the following equality and comparing the result with (2) in Step 1 while regarding variable x in the equality below to be x_j in (2):

$$F(x)=(((F(x_1)(x-x_2)(x-x_3)\ldots(x-x_k)/(x_1-x_2)(x_1-x_3)\ldots(x_1-x_k)\ldots\ldots.. + F(x_k)(x-x_1)(x-x_2)\ldots(x-x_{k-1})/(x_k-x_1)(x_k-x_2)\ldots..))) \bmod p$$

Step 3 in the above algorithm is additionally included for the purpose of computing the values of parameters c_i in the pro- posed method. In other applications, if only the secret value d need be recovered, this step may be eliminated.

III. Image Authentication and Data Repairing

In the proposed method, a PNG image is created from a binary-type grayscale document image I with an alpha channel plane. The original image I may be thought as a grayscale channel plane of the PNG image. An illustration of this process of PNG image creation is shown in fig. 2. Next I, is binarized by moment-preserving thresholding [13], yielding a bi- nary version of I, which we denote as I_b. Data for authentication and repairing are then computed from I_b and taken as input to the Shamir secret sharing scheme to generate n secret shares. The share values are subsequently mapped into a small range alpha channel values near the maximum transparency value to create an imperceptibility effect. Finally, the mapped secret shares are randomly embedded into the alpha channel for the purpose of promoting the security protection and data repair capabilities. Two block diagrams describing the proposed method are shown in figs. 3 and 4.

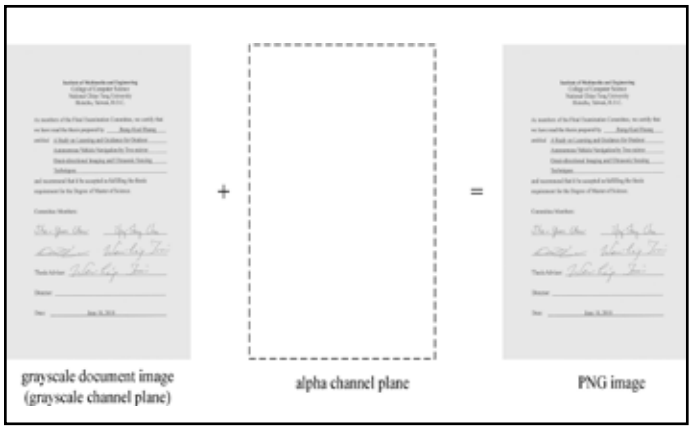


Fig. 2: Illustration of Creation of a PNG Image from a Grayscale Document Image and an Additional Alpha Channel Plane

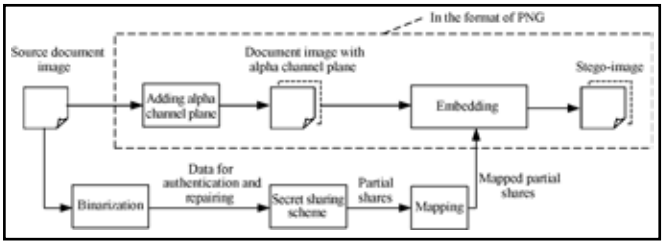


Fig. 3: Illustration of Creating a PNG Image From a Grayscale Document Image and an Alpha Channel

Algorithm 3: Generation of a stego-image in the PNG format from a given grayscale image.
Input: a grayscale document image I with two major gray values and a secret key$K$.
Output: stego-image I' in the PNG format with relevant data embedded, including the authentication signals and the data used for repairing.

Steps.
Stage I—generation of authentication signals.
Step 1. (Input image binarization) Apply moment-preserving thresholding [13] to I to obtain two representative gray values g_1 and g_2 , compute $T=(g_1+g_2)/2$, and use$T$ as a threshold to binarize I, yielding a binary version I_b with "0" representing g_1 and "1" representing g_2 .
Step 2. (Transforming the cover image into the PNG format) Transform I into a PNG image with an alpha channel plane I_α by creating a new image layer with 100% opacity and no color as I_α and combining it with I using an image processing software package.
Step 3: Take in an unprocessed raster-scan order a 2 × 3 block Bb of Ib of with pixels p_1, p_2……p_6.
Step 4. (Creation of authentication signals) Generate a 2-bit authentication signal s = a1a2 with a1=$p_1 \oplus p_2 \oplus p_3$ and a2=$p_4 \oplus p_5 \oplus p_6$, where $\oplus$ denotes the exclusive-or operation.
Stage II—creation and embedding of shares.
Step 5. (Creation of data for secret sharing) Concatenate the 8 bits of a_1, a_2 , and p_1 through p_6 to form an 8- bit string, divide the string into two 4-bit segments, and transform the segments into two decimal numbers m_1 and m_2 , respectively.
Step 6. (Partial share generation) Set p, c_i, and x_i in (1) of Algorithm 1 to be the following values: 1) p=17 (the smallest prime number larger than 15); 2) $d=m_1$ and $c_1=m_2$; and 3) $x_1=1$, $x_2=2$…$x_6=6$. Perform Algorithm 1 as a (2, 6) -threshold secret sharing scheme to generate six partial shares q_1 through q_6 using the following equations:

$$q_i = F(x_i) = (d+c_{1xi}) \bmod p \quad (3)$$

where i = 1,2,……6.
Step 7. Add 238 to each of q1 to q6 then the new resulting values are q'1 through q'6 respectively, which fall in the nearly total transparency range of 238 through 254 in the alpha channel plane I_α.
Step 8. (Embedding two partial shares in the current block) Take block B_α in I_α corresponding to B_b in I_b, select the first two pixels in B_α in the raster-scan order, and replace their values by q'1 and q'2 respectively.
Step 9. (Embedding remaining partial shares at random pixels) Use key K to select randomly four pixels in I_α but outside B_α, which are unselected yet in this step, and not the first two pixels of any block; in the raster-scan order, replace the four pixels values by the remaining four partial shares q'3 through q'6
Step 10. (End of looping) If there exists any unprocessed block in I_b, then go to Step 3; otherwise, take the final I in the PNG format as the desired stego-image I'.
The reason why we choose the prime number to be 17 in the above algorithm is explained here. If it was instead chosen to be larger than 17, then the aforementioned interval will be en- larged, and the values of q'_1 through q'_6 will become possibly smaller than 238, creating an undesired less transparent but vi- sually whiter stego-image. On the other hand, the 8 bits men- tioned in Steps 5 and 6 above are transformed into two decimal numbers m1 and m2 with their maximum values being 15 (see Step 5 above), which are constrained to lie in the range of 0 through p-1. Therefore, p should not be chosen to be smaller than 16. In short is an optimal choice.
As to the choice of the block size, the use of a larger block size, such as 2 × 4 or 3 × 3 will reduce the precision of the resulting integrity authentication (i.e., the stego-image will be verified in a spatially coarser manner). On the other hand, it seems that a smaller block size such as 2 × 2 instead of 2 × 3 may be tried to increase the authentication precision. However, a block in the alpha channel with a size of 2 × 2 can be used to embed only four partial shares instead of six (see Steps 6–9 of Algorithm 3). This decreases the share multiplicity and thus reduces the data repair capability of the method. In short, there is a tradeoff between the authentication precision and the data repair capability, and our choice of the block size of 2 × 3 is a balance in this aspect.

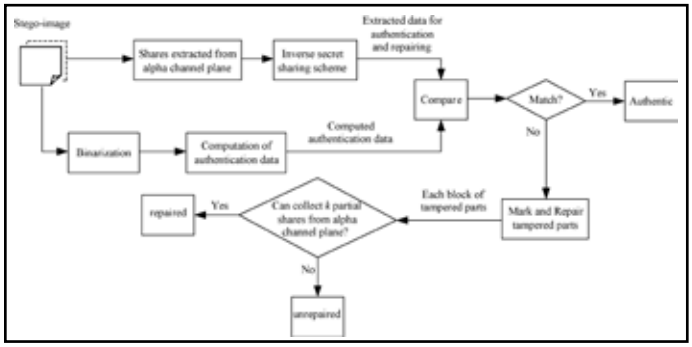


Fig. 4: Authentication Process Including Verification and Self-Repairing of a Stego-Image in PNG Format

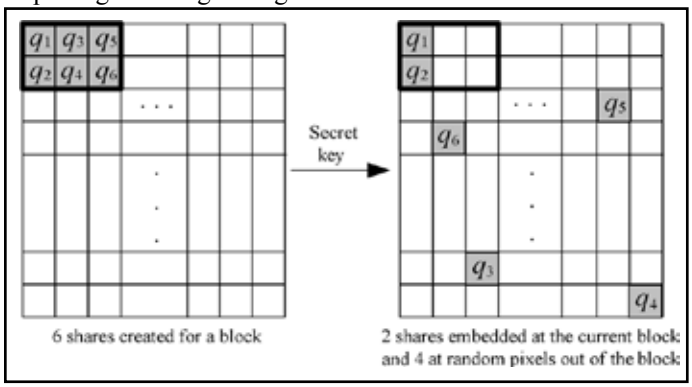


Fig. 5: Illustration of Embedding Six Shares Created for a Block: Two Shares Embedded at the Current Block, and the Other Four in Four Randomly Selected Pixels Outside the Block, With Each Selected Pixel Not Being the First Two Ones in Any Block

Algorithm 4: Authentication of a given stego-image in the PNG format.
Input: stego-image I', the representative gray values g_1 and g_2 , and the secret key K used in Algorithm 3.
Output: image I_r with tampered blocks marked and their data repaired if possible.

Stage I: extraction of the embedded two representative gray values.

Step 1. (Binarization of the stego-image) Compute $T=(g_1+g_2)/2$, and use it as a threshold to binarize I', yielding a binary version I'$_b$ with "0" representing g_1 and "1" representing g_2.

Stage II: verification of the stego-image.

Step 2. (Beginning of looping) Take in a raster scan order an unprocessed block B'$_b$ from I'$_b$ with pixel values p_1 through p_6, and find six pixels values q'$_1$ through q'$_6$ of the corresponding block B'$_\alpha$ in the alpha channel plane I'$_\alpha$ of I'.

Step 3. (Extraction of the hidden authentication signal) Perform the following steps to extract the hidden 2-bit authentication signal $s=a_1a_2$ from B'_α .

(1) Subtract 238 from each of q'$_1$ and q'_2 to obtain two partial shares q_1 and q_2 of B'$_b$, respectively.

(2) With shares (1, q_1) and (2, q_2) as input, perform Algorithm 2 to extract the two values d and c_1 (the secret and the first coefficient value, respectively) as output.

(3) Transform d and c_1 into two 4-bit binary values, concatenate them to form an 8-bit string S,and take the first 2 bits a_1 and a_2 of S to compose the hidden authentication signal s = a_1a_2 .

Step 4. (Computation of the authentication signal from the current block content) Compute a 2-bit authentication signal s'=a'1a'2 from values p_1 through p_6 of the six pixels of B'$_b$ by a'$_1$ = $p_1 \oplus p_2 \oplus p_3$ and a'$_2$ = $p_4 \oplus p_5 \oplus p_6$

Step 5. (Matching of the hidden and computed authentication signals and marking of tampered blocks) Match s and s' by checking if $a_1=a'_1$ and $a_2=a'_2$, and if any mismatch occurs, mark B'b , the corresponding block B' in I', and all the partial shares embedded in $B\acute{\alpha}$ as tampered.

Step 6. (End of looping) If there exists any unprocessed block in I', then go to Step 2; otherwise, continue.

Stage III: self-repairing of the original image content

Step 7. (Extraction of the remaining partial shares) For each block B'$_\alpha$ in I'$_\alpha$, perform the following steps to extract the remaining four partial shares q_3 through q_6 of the corresponding block B'$_b$ in I'$_\alpha$ from blocks in other than B'$_\alpha$.

(1) Use key K to collect the four pixels in I'$_\alpha$ in the same order as they were randomly selected for B'$_b$ in Step 9 of Algorithm 3, and take out the respective data q'$_3$, q'$_4$, q'$_5$, and q'$_6$ embedded in them.

(2) Substract 238 from each of q'$_3$ through q'$_6$ to obtain q_3 through q_6, respectively.

Step 8. (Repairing the tampered regions) For each block B' in I' marked as tampered previously, perform the following steps to repair it if possible.

(1) From the six partial shares q_1 through q_6 of block B'$_b$ in I'$_b$ corresponding to B' choose two of them, e.g q_k and q_l which are not marked as tampered, if possible.

(2) With shares (k, q_k),(l, q_l)perform algorithm 2 to extract d and c_1 as output.

(3) Transform d and c_1 into two 4 bit binary values and concatenate to form 8 bit string s'.

(4) Take last 6 bits b'$_1$,b'$_2$….b'$_6$ from S', and check their corresponding tampered pixel values y'$_1$, y'$_2$…y'$_6$ by the following way.

if b'$_i$=0, set y'$_i$=g_1;otherwise, set y'i = g_2

where i = 1,2,…..6.

Step 9. Take the final I' as the desired self-repaired imageI_r.

IV. Conclusion

The generated authentication signal and the content of a block have been transformed into partial shares by the Shamir method, which have been then distributed in a well-designed manner into an alpha channel plane to create a stego-image in the PNG format. The undesired opaque effect visible in the stego-image coming from embedding the partial shares has been eliminated by mapping the share values into a small range of alpha channel values near their maximum transparency value of 255.

References

[1] C. S. Lu, H. Y. M. Liao,"Multipurpose watermarking for image authentication and protection", IEEE Trans. Image Process., Vol. 10, No. 10, pp. 1579–1592, Oct. 2001.

[2] M. U. Celik, G. Sharma, E. Saber, A. M. Tekalp, "Hierarchical watermarking for secure image authentication with localization", IEEE Trans. Image Process., Vol. 11, No. 6, pp. 585–595, Jun. 2002.

[3] Z. M. Lu, D. G. Xu, S. H. Sun,"Multipurpose image watermarking algorithm based on multistage vector quantization", IEEE Trans. Image Process., Vol. 14, No. 6, pp. 822–831, Jun. 2005.

[4] M. Wu, B. Liu,"Data hiding in binary images for authentication and annotation", IEEE Trans. Multimedia, Vol. 6, No. 4, pp. 528–538, Aug. 2004.

[5] H. Yang, A. C. Kot,"Binary image authentication with tampering localization by embedding cryptographic signature and block identifier", IEEE Signal Process. Lett., Vol. 13, No. 12, pp. 741–744, Dec. 2006.

[6] H. Yang, A. C. Kot,"Pattern-based data hiding for binary images authentication by connectivity-preserving", IEEE Trans. Multimedia, Vol. 9, No. 3, pp. 475–486, Apr. 2007.

[7] H. Y. Kim, A. A?,"Secure authentication watermarking for halftone and binary images", Int. J. Imag. Syst. Technol., Vol. 14, No. 4, pp. 147–152, 2004.

[8] C. H. Tzeng, W. H. Tsai,"A new approach to authentication of binary images for multimedia communication with distortion reduction and security enhancement", IEEE Commun. Lett., Vol. 7, No. 9, pp. 443–445, Sep. 2003.

[9] Y. Lee, J. Hur, H. Kim, Y. Park, H. Yoon,"A new binary image authentication scheme with small distortion and low false negative rates", IEICE Trans. Commun., Vol. E90-B, No. 11, pp. 3259–3262, Nov. 2007.

[10] Y. Lee, H. Kim, Y. Park,"A new data hiding scheme for binary image authentication with small image distortion", Inf. Sci., Vol. 179, No. 22, pp. 3866–3884, Nov. 2009.

[11] A. Shamir,"How to share a secret", Commun. ACM, Vol. 22, No. 11, pp. 612–613, Nov. 1979.

[12] C. C. Lin, W. H. Tsai,"Secret image sharing with steganography and authentication", J. Syst. Softw., Vol. 73, No. 3, pp. 405–414, Nov./ Dec. 2004.

[13] W. H. Tsai,"Moment-preserving thresholding: A new approach", Comput. Vis. Graph. Image Process., Vol. 29, No. 3, pp. 377–393, Mar. 1985.

ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

T. Shailendra a student of M.Tech in Computer Science from Sir C.R. Reddy College of Engineering, Eluru, WestGodavari Dt, Andhra Pradesh, India.

M.Madhava Rao is a Research Scholar in CSE Depart ment Aacharya Nagarjuna University. He completed his M.Tech in CSE, Aacharya NagarjunaUniversity Campus .He is working as Asst. Professor in C R REDDY College of Engineering, Eluru, Andra Pradesh.He is having about 8 years of teaching experience in different Engineering Colleges.