International Journal of Engineering Research and General Science Volume 3, Issue 3, May-June, 2015
ISSN 2091-2730

# Secrete Sharing Based Method for Ensuring Authenticity of Gray Scale Document Images

Reshma D. Vartak, Smita Deshmukh

Department of Information Technology, Terna Engineering College, Nerul, Navi Mumbai, India,

**Email**-reshma.vartak@gmail.com

**Abstract**-Digital image authentication is difficult for binary images because of its simple binary nature which leads to visible changes after authentication signals are embedded in the image pixels. Hence effective solution to ensure authentication of binary images should take in to account not only security issue of preventing image tampering but also keeping visual quality of resulting image. A new authentication method for binary like gray scale document images based on the secret sharing technique with a data repair capability via the use of the Portable Network Graphics (PNG) image is proposed. An authentication signal is generated for each block of a gray scale document image, which together with the binaries image block content, is transformed into several shares using the Shamir secret sharing scheme and then these shares are embedded into an alpha channel plane. Alpha channel plane is transparent plane; this plane is then combined with the original gray scale image to form stego image in PNG format. In the process of image authentication and verification, an image block is marked as tampered if the authentication signal computed from the current image block content does not match with shares extracted which are embedded in the alpha channel plane. Data repairing is then applied to each tampered image block by a reverse Shamir scheme after collecting two shares from unmarked blocks. Measures for protecting the security of the image content hidden in the alpha channel plane are also proposed.

**Keywords**-Gray Scale image authentication, data hiding, alpha channel plane, data repair, portable network graphics, secrete sharing, pixel by pixel repair.

## 1. INTRODUCTION

Digital images are used to preserve important information. But providing integrity and authentication to these images is a challenging task as they are increasingly transmitted over insecure network such as internet. In this era with the use of fast advanced technologies it is easy to modify the contents of these digital images. Therefore there is need to protect these images against various attempts to manipulate them and it is important to make an effective method to solve image authentication problem that is ensuring the integrity of an image [1] [2], particularly for document images such as important certificates, scanned cheques, art drawings, signed documents, circuit diagrams, design drafts etc.

Image authentication is difficult for gray scale and binary document images, because of their simple binary nature that leads to perceptible changes after authentication signal are embedded in the image pixels [3] [4]. Many conventional methods have been proposed for authentication of gray scale document images [8] [9] but there is no pixel by pixel data repair capability.

So in this paper we proposed a new gray scale document image authentication method with an additional self-repair capability for fixing tampered image data. The input cover image is assumed to be a binary-like gray scale image with two major gray values, one

being background and other being foreground. After the proposed method is applied, the input cover image is transformed into a *stego-image* in the Portable Network Graphics (PNG) format with an additional alpha channel that carries authentication data and original image data 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 resulting image is regarded as inauthentic, meaning that the fidelity check of the image fails. The proposed method is based on (k, n) threshold secret sharing scheme proposed by Shamir[7], in which a secret image is divided into *shares* for keeping by participants, and when sufficient numbers of shares are collected, the secret message can be losslessly recovered.

Proposed method is divided in to two parts; ***Generating stego image from input cover image*** *and* ***image authentication and repairing of received or retrieved stego image***.

**Shamir Method for Creating Secret Shares**

In the (k, n)-threshold secret sharing method proposed by Shamir[7] ,secret d in the form of an integer is transformed into shares, which then are distributed to participants for them to keep; and as long as of the shares are collected, the original secret can be accordingly recovered, where k<=n.

**Algorithm 1: for (k, n) -Threshold Secret Sharing**

***Input***: d secret in the form of an integer, number of participants n, and threshold k<=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.

**Step 2:** Select k-1 integer values c1, c2… ck-1 within the range of 0 through p-1.

**Step 3:** Select n distinct real values x1, x2, ..,xn.

**Step4**: Use the following (k-1)-degree polynomial to compute n function values F(xi) , called *partial shares* for i=1,2,3… , n i.e.,

$$F(x_i) = \left(d + c_1 x_i + c_2 x_i^2 + \cdots + c_{k-1} x_i^{k-1}\right)_{\mathrm{mod} p} . \qquad (1)$$

**Step 5:** Deliver the two-tuple (xi, F (xi)) as a *share* to the ith participant where i=1, 2, 3…, n.

Since there are k coefficients, namely d and c1and through ck-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 coefficients in order to recover secret d.

**Algorithm 2: for Secret Recovery**

***Input:*** k shares collected from the participants and the prime number p

***Output*:** secret d hidden in the shares and coefficients ci used in (1) in Algorithm 1, where i= 1, 2, …., k-1.

**Step 1:** Use the k shares,

(x1, F(x1)), (x2, F(x2)), … , (xk, F(xk)) to set up

$$F(x_j) = \left(d + c_1 x_j + c_2 x_j^2 + \cdots + c_{k-1} x_j^{k-1}\right)_{\mathrm{mod}p} \qquad (2)$$

**Step 2:** Solve the equations above by Lagrange's interpolation to obtain as follows

$$d = (-1)^{k-1}\left[F(x_1)\frac{x_2 x_3 \ldots x_k}{(x_1-x_2)(x_1-x_3)\ldots(x_1-x_k)} + F(x_2)\frac{x_1 x_3 \ldots x_k}{(x_2-x_1)(x_2-x_3)\ldots(x_2-x_k)} + \cdots + F(x_k)\frac{x_1 x_2 \ldots x_{k-1}}{(x_k-x_1)(x_k-x_2)\ldots(x_k-x_{k-1})}\right]_{\mathrm{mod}p}.$$

**Step 3:** Compute c1 through ck-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 xj in (2):

$$F(x) = \left[F(x_1)\frac{(x-x_2)(x-x_3)\ldots(x-x_k)}{(x_1-x_2)(x_1-x_3)\ldots(x_1-x_k)} + F(x_2)\frac{(x-x_1)(x-x_3)\ldots(x-x_k)}{(x_2-x_1)(x_2-x_3)\ldots(x_2-x_k)} + \cdots + F(x_k)\frac{(x-x_1)(x-x_2)\ldots(x-x_{k-1})}{(x_k-x_1)(x_k-x_2)\ldots(x_k-x_{k-1})}\right]_{\mathrm{mod}p}.$$

## 2. PROPOSED METHOD

### 2.1 Generation of stego image in PNG format from gray scale document image:

Cover gray scale document image E is converted into binary image Eb by using moment preserving threshold. This *E*b is taken as an input to Shamir's secret sharing scheme to generate n secret shares. Cover gray scale image is combined with alpha channel plane $E\alpha$ using image processing software to obtain PNG image. Using binary image Eb authentication data is generated for each 2*3 block and which is then combined with original image data; six shares are generated for each 2*3 image block using Shamir secrete sharing. The resulting shares are embedded on to the alpha channel plane of PNG image using chaotic logistic to obtain Stego image *E*'. Figure 1 illustrates creation of PNG image and figure 2 illustrates block diagram of generating stego image from given gray scale document image. Share embedding process is illustrate in figure3.

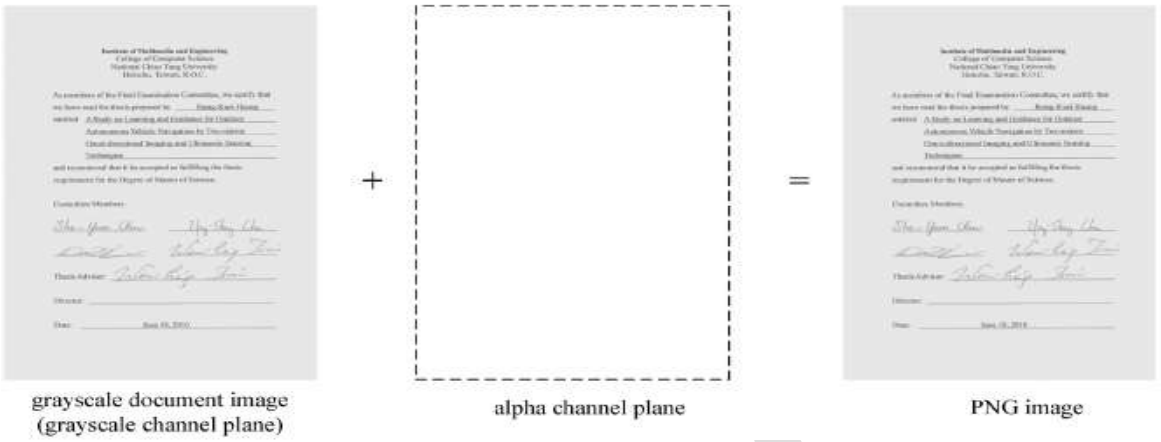



**Figure 2.1 Creation of PNG Image from Gray scale Document Image and Alpha channel plane**

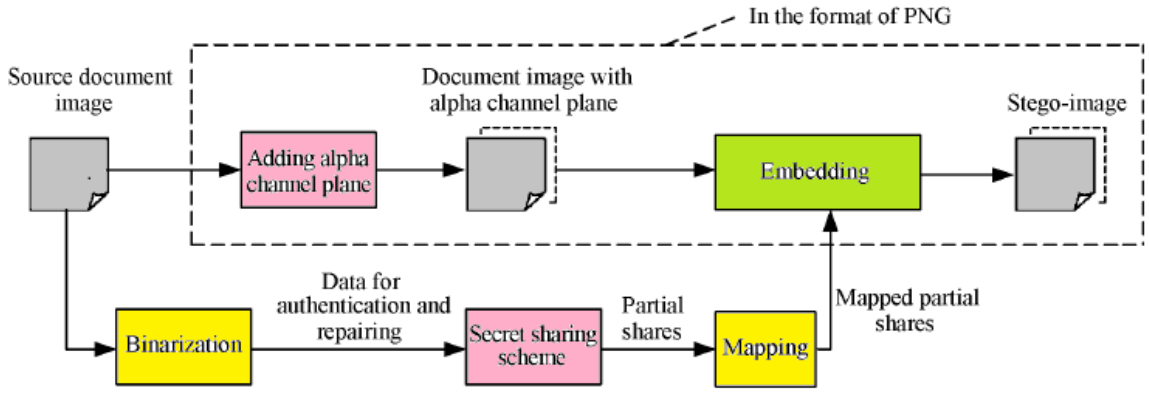



**Figure 2.2 Generating Stego- image in PNG Format from Gray scale Document Image**

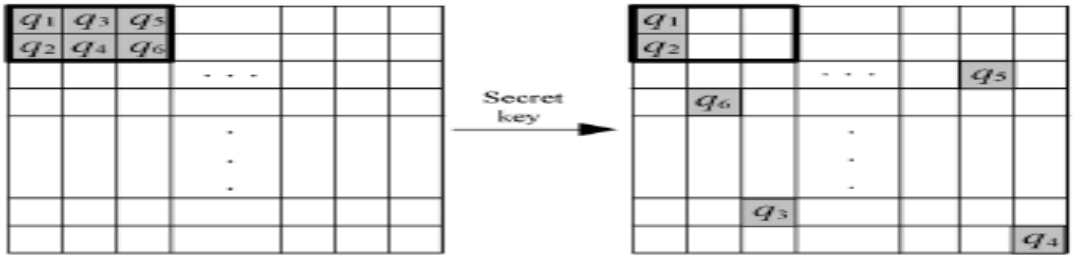



**Figure 2.3 Pictorial representation of embedding 6 shares generated for a 2*3 image block, 2 shares embedded in current block and other 4 in 4 randomly selected pixels outside the block, with each selected pixel not being the first 2 one in any block**

### 2.1.1 Algorithm 3: for Generating Stego- image in PNG Format from a given Gray scale Image.

***Input:*** A gray scale document image E with two major gray values and secret key K.

***Output:*** Stego image $E'$ in PNG with encrypted format, relevant data embedded, including the authentication signal and the data used for repairing.

- **Part 1: Authentication signal generation**

**Step 1**: **Binarization of input image**

Moment preserving threshold [3] applied to E to obtain two representative gray values $g_1$ and $g_2$. Computing the average of $g_1$ and $g_2$ to obtain the threshold value. Use this threshold to binarize E, yielding a binary version of $E_b$ with "0" representing $g_1$ and "1" representing g2.

**Step 2: Conversion of cover image into PNG format**

Convert E into PNG image with an alpha channel plane $E\alpha$ by creating new image layer with 100% opacity and no color as $E\alpha$ and combining it with E using an image processing software package.

**Step 3: Starting of loop**

Take in an unrefined raster scan order of 2*3 block $B_b$ in $E_b$ with pixels $p_1$ , $p_2$…..p6.

**Step 4: Authentication signal generation**

Generate 2-bit authentication signal

$L=b_1b_2$ with $b_1=p_1 \oplus p_2 \oplus p_3$ and $b_2=p_4 \oplus p_5 \oplus p_6$.

- **Part 2: Design and embedding of shares**

**Step 5: Creation of data for secret sharing**

Concatenate the 8 bits of $b_1$, $b_2$ and $p_1$ through $p_6$ form an 8-bit string, divide this string into two 4-bit segments, and convert the segment into 2 decimal numbers $a_1$ and $a_2$ respectively.

**Step 6: Generation of partial shares**

Set p, $m_j$, and $y_j$ in eqn. (1) of Algorithm 1 to,

1) p=17 (the smallest Prime number larger than 15);

2) c=$\alpha_1$ and $m_{1=}\alpha_2$; and

3) $y_1=1, y_2=2, \ldots\ldots y_6=6$.

Perform algorithm 1 as a (2, 6) threshold secret sharing scheme and generate six partial shares $r_1$ through $r_6$ using the following equations:

$$r_j = F(y_j) = (c + m_1 y_j) \bmod p \qquad \textbf{(3)}$$

Where j= 1, 2, …. , 6

**Step 7: Mapping of partial shares**

Add 238 to each of $r_1$ through $r_6$ , resulting in the new value of $r'_1$ through $r'_6$ respectively, which fall in the nearly total transparency range of 238 through 254 in the alpha channel plane $E\alpha$.

**Step 8: Embedding two fractional shares in the current block**

Take block $B\alpha$ in $E\alpha$ corresponding to $B_b$ in $E_b$, select the first two pixels in $B\alpha$ in the raster scan order and replace their values by $r'_1$ and $r'_2$ respectively.

**Step 9: Embedding remaining partial shares at random pixels**

Use key K to select randomly four pixels in $E\alpha$ but outside $B\alpha$, not the first two pixels of any block; in the raster scan order, and replace four pixels values by the remaining four partial shares $r'_3$ through $r'_6$ generated above, respectively.

**Step 10: End of loop**

If there exist any unprocessed block in $E_b$, then go to step 3 otherwise take the E in the PNG format.

- **Part 3: PNG image encryption**

**Step 11: Encryption of the PNG image**

Encrypt the PNG image using chaotic logistic map, take the final E in PNG with encrypted format as the desired stego-image $E'$.

The prime number p used here is 17, so the values of $r_1$ through $r_6$ yield by equation (3) are between 0 and 16.After executing step 7 of above algorithm, they become $r'_1$ and $r'_2$ respectively. Which all fall into the small interval of integers ranging from 238 to 254. Consequent embedding of $r'_1$ through $r'_2$ in a narrow interval into the alpha channel plane means that very alike values will appear everywhere in the plane, resulting in a nearly uniform transparency effect, which will not stimulate notice from an attacker. We choose prime number to be17 in the above algorithm because, if it was chosen instead to be larger than 17, then the above mentioned interval will be enlarged and the values of $r'_1$ through $r'_6$ will become possibly smaller than 238, creating visually whiter stego image. In contrast, the 8 bits mentioned in steps 5 and 6 above are transformed into two decimal numbers $b_1$ and $b_2$ with their maximum values being 15(step 5 above) , which are forced to lie in the range of 0 through p-1 (step 2 in algorithm 1). Therefore p p=17 is the best possible answer.

## 2.2 Stego- image Authentication and Data Repairing

Stego-image in PNG format, when received or retrieved is decrypted and then authenticated by the proposed method for its authenticity and integrity modifications. Integrity modifications of the image are identified at block levels and prepared at pixel levels. Figure 4 and figure 5 illustrates block diagram of stego image authentication and data repairing process of tampered image block.

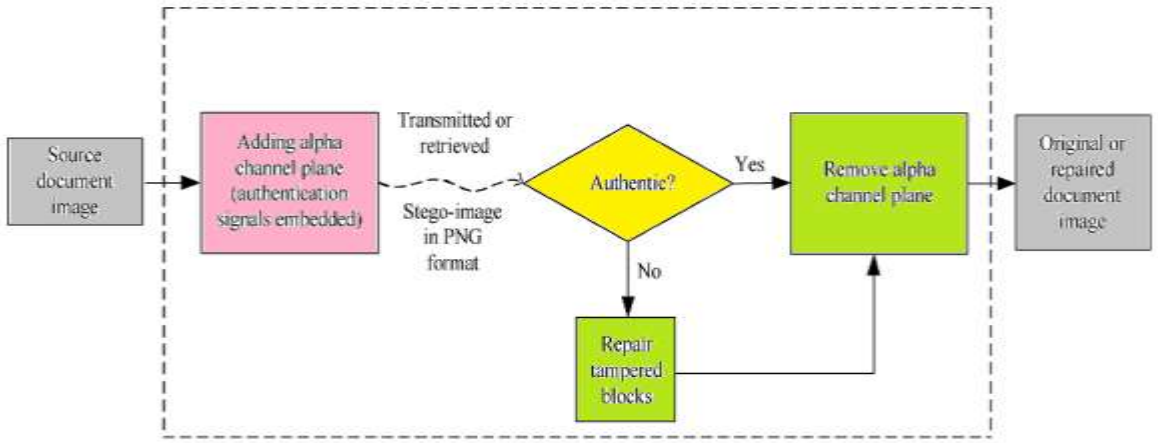



**Figure 2.4 Proposed Document Image Authentication Process**

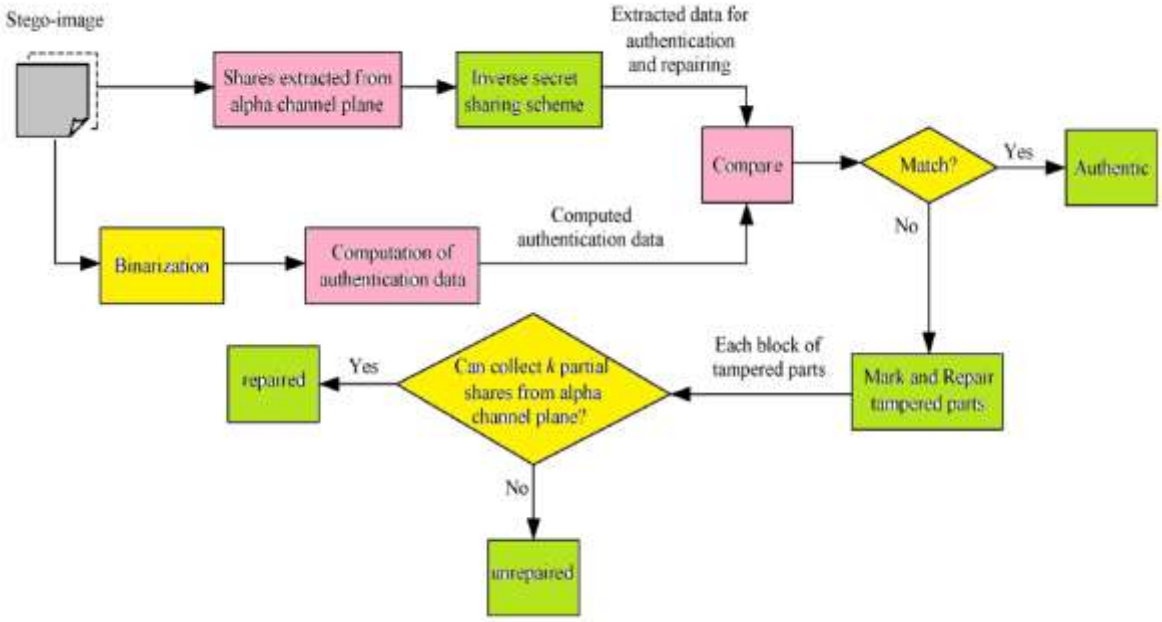



**Figure 2.5 Authentication Process including Verification and Self-repairing of a Stego-image in PNG Format**

### 2.2.1 Algorithm 4: for Stego-Image Authentication

***Input:*** A stego image $E'$ with gray values $g_1$ and $g_2$ and secret key K used in algorithm 3.

***Output:*** Image $E_r$ with tampered blocks marked and their data repaired if possible.

- **Part 1: Decryption of stego image and extraction of the two representative gray values.**

**Step 1: Decryption of stego image**

Decrypt the stego image by the random key used in the encryption.

**Step 2: Conversion of decrypted image into binary form**

Compute T= $(g_1 + g_2)/2$, using this threshold value to Convert $E'$ into binary form $E'_b$ with "0" representing $g_1$ and "1" representing $g_2$.

- **Part 2: Stego image authentication**

**Step 3: Start loop**

Take in a raster scan order an unprocessed block $B'_b$ from $E'_b$ with pixel values $p_1$ through $p_6$ and find six pixel values $r'_1$ through $r'_6$ of the corresponding block $B\alpha'$ in the alpha channel plane $E\alpha'$ of $E'$.

**Step 4: Drawing out of secret authentication signal**

The following steps perform to extract the hidden 2-bit authentication signal $L=b_1b_2$ from $B\alpha'$.

1) Subs tract 238 from each of $r'_1$ and $r'_2$ to obtain 2 partial shares $r_1$ and $r_2$ of $B'\alpha$, respectively.
2) With shares (1, $r_1$) and (2, $r_2$) as input, perform Algorithm 2 to extract the two values c and $m_1$ (secret and first coefficient value) as output.
3) Transform c and $m_1$ into two 4 bit binary values , concatenate them to form an 8-bit string S, and take the first 2 bits of S to compose the hidden authentication signal $L=b_1b_2$.

**Step 5: Computation of authentication signal from the current block content**

Compute 2 bit authentication signal $L'=b'_1b'_2$ from values $p_1$ through $p_6$ of six pixels of $B'_b$ by $b'1 = p_1 \oplus p_2 \oplus p_3$ and $b'_2 = p_4 \oplus p_5 \oplus p_6$.

**Step 6: Comparison of computed authentication signal with hidden shares and marking the tampered block**

Matching L and $L'$ by checking if $b_1=b'_2$ and $b_2=b'_2$ and if any mismatch occurs mark $B'_b$, the corresponding block $B'$ in $E'$ and all the partial shares embedded in $B\alpha'$as tampered.

**Step 7: End loop**

If there exist any unprocessed block in $E'_b$ then go to step 3, otherwise continue.

- **Part 3: Self-repairing of the original image content**

**Step 8: Drawing out of the remaining partial shares**

For each block $B\alpha'$ in $E\alpha'$ , execute the following step to extract the remaining 4 partial shares $r_3$ through $r_6$ of the corresponding block $B'_a$ in $E'\alpha$ from blocks in $E\alpha'$ other than $B\alpha'$ .

1) Use key K to collect the four pixels in $E\alpha'$ in the same order as they were randomly selected for $B'_b$ in step 9 of the algorithm 3, and take out the respective data r'3, r'4, r'5, r'6 embedded in them.
2) Subtract 238 from each of $r'_3$ through $r'_6$ to obtain $r_3$ through $r_6$, respectively.

**Step 9: Repair the tampered regions**

For each block $B'$ in $E'$ marked as tampered previously, execute the following steps to repair if it possible.

1) From the six partial shares $r_1$ through $r_6$ of block $B'_b$ in $E'_b$ corresponding to $B'$ (two computed in step 4(1) and four in step 8(2) above), choose two of them, say $r_k$ and $r_j$ which are not marked as tampered , if possible.

2) With shares (k, $r_k$ ) and (l, $r_l$ ) as input , perform Algorithm 2 to extract the values of c and $m_1$ (secret and first coefficient value)as output.

3) Transform c and $m_1$ into two 4 bit binary values, and concatenate these 2 binary values to form an 8-bit string ′

4) Take the last six bits $a'_1$ , $a'_2$ , … , $a'_6$ from $S'$ , and check their binary values to repair the corresponding tampered pixel values $z'_1$ ,$z'_2$ , … ,$z'_6$ of block $B'$ by the following way: if $a'_j$ =0, set $z_j = g_1$ ; otherwise set $z'_j = g_2$ ; where j=1,2…6.

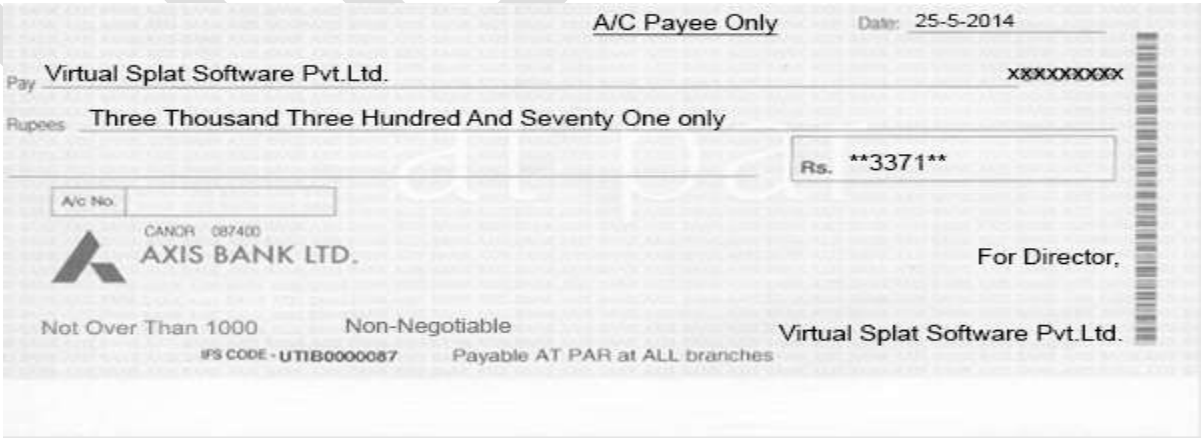
**Step10. Take the final $E'$ as the desired self –repaired image $E_r$.**

## 3. RESULTS AND DISCUSSIONS

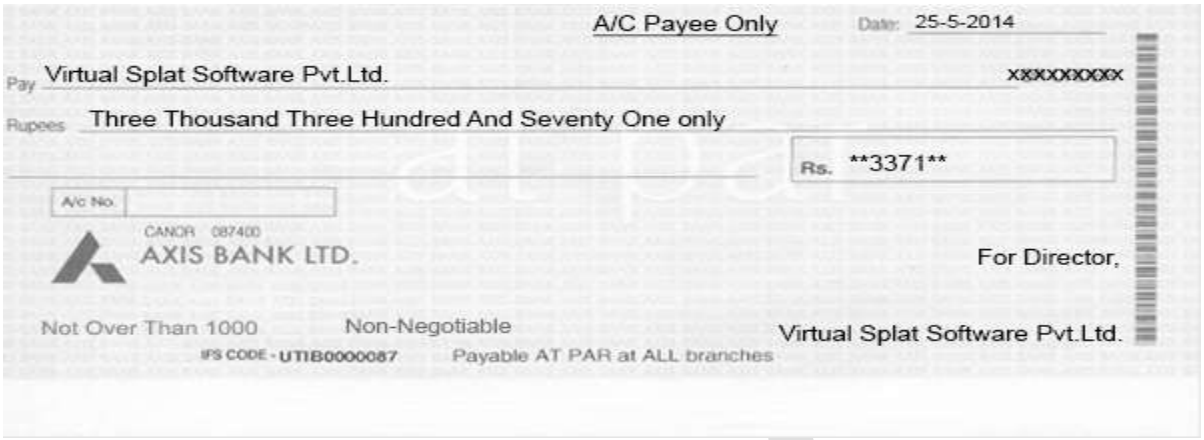
### 3.1 Experimental Results

The experimental results of proposed method using gray scale document images of bank checks are shown in figure 3.1, figure 3.2, and figure 3.3.

Figure 3.1 shows results of applying proposed method using a gray scale document image of a check; *figure 3.1(a) shows original cover image*, the result of applying Algorithm 3 to embed data for authentication and data for repairing is shown in *figure 3.1 (b) as a stego-image with embedded data* which is visually almost identical to original cover image, *figure 3.1(c) shows tampered image* which is actually generated stego-image tampered using superimposing a rectangular block of background color and replacing original text by fake text using paint application, *figure 3.1(d) shows extracted image* with black dots indicating unrepaired image pixels , exacted image is result of applying Algorithm 4 to detect image tampering and recover original image using data embedded in alpha channel plane.

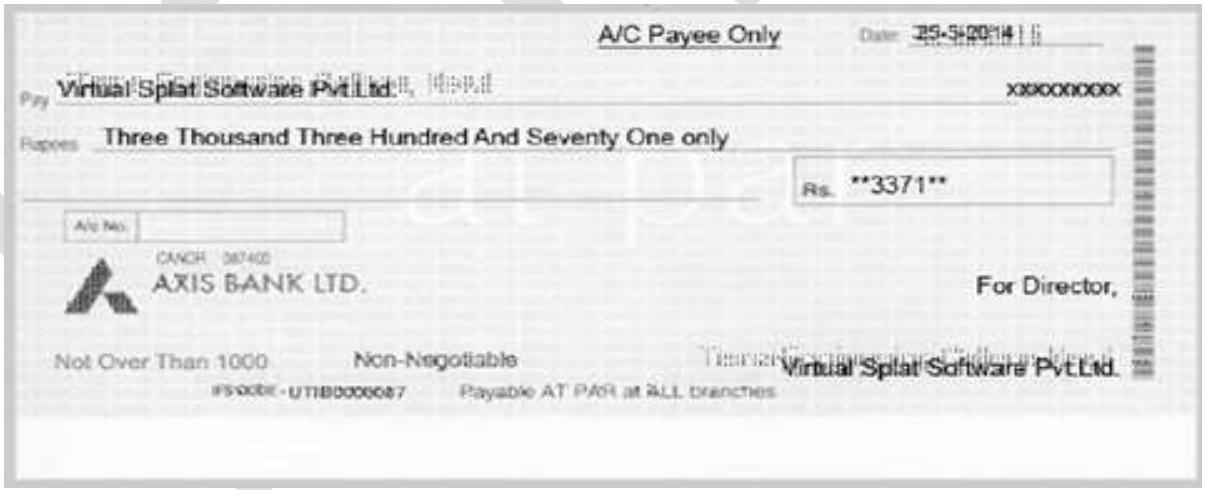



**(a) Original Cover Image**




**(b) Stego- Image with embedded data**


**(c) Tampered Image**



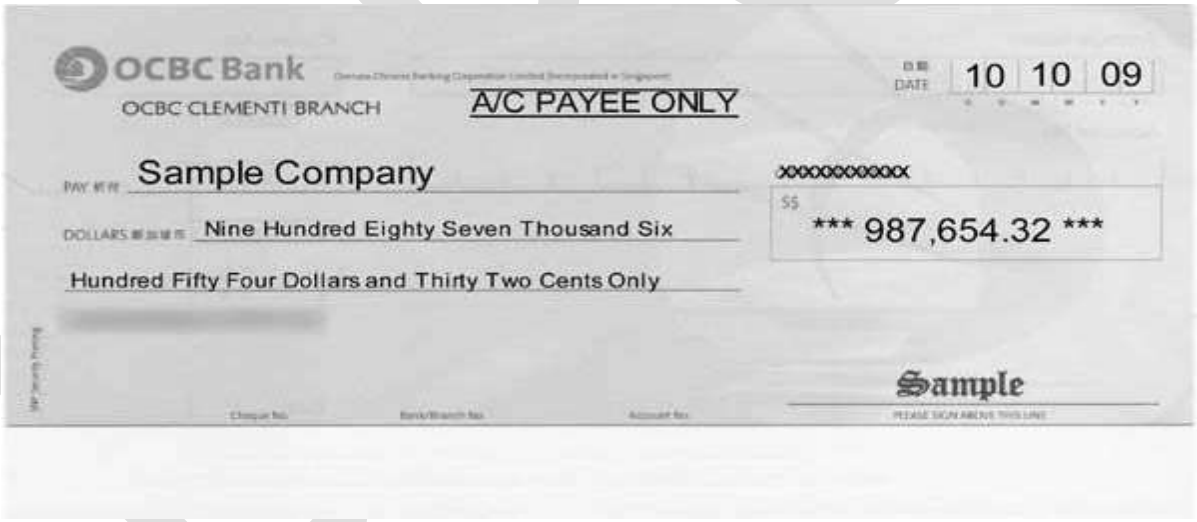

**(d) Extracted Image**

**Figure 3.1 Experimental result of a document image of a bank check attacked by superimposing a rectangular shape of background color on a piece of text and replacing original text by fake text**

(a) Original Cover Image ; (b) Stego-image with embedded data;
(c ) Tampered Image yielded by superimposing operation; (d) Extracted Image with dots indicating unrepaired tampered image pixels

Figure 3.2 shows results of applying proposed method using a gray scale document image of a check; figure 3.2(a) *shows original cover image*, the result of applying Algorithm 3 to embed data for authentication and data for repairing is shown in *figure* 3.2 *(b) as a stego-image with embedded data* which is visually almost identical to original cover image, *figure* 3.2 *(c) shows tampered image* which is actually generated stego-image tampered by superimposing rectangular shape of background color on original image data using paint application, *figure* 3.2 *(d) shows extracted image* with black dots indicating unrepaired image pixels, extracted image is result of applying Algorithm 4 to detect image tampering and recover original image using data embedded in alpha channel plane.


**(a) Original Cover Image**




**(b) Stego -Image with embedded data**


**(c ) Tampered Image**


**(d ) Extracted Image**

**Figure 3.2 Experimental result of a document image of a bank check attacked by superimposing a rectangular shape of background color on a piece of text**

(a ) Original Cover Image ;(b) Stego-image with embedded data;
(c ) Tampered Image yielded by superimposing operation; (d ) Extracted Image with black dots indicating unrepaired tampered image pixels

Figure 3.3 shows results of applying proposed method using a gray scale document image of a check; *figure 3.3 (a) shows original cover image*, the result of applying Algorithm 3 to embed data for authentication and data for repairing is shown in *figure 3.3 (b) as a stego-image with embedded data* which is visually almost identical to original cover image, *figure 3.3(c) shows tampered image* which is actually generated stego-image tampered by erasing original image data using paint application, *figure 3.3(d) shows erroneous extracted image* obtained with wrong key used for repairing, extracted image is result of applying Algorithm 4 to detect image tampering and recover original image using data embedded in alpha channel plane.


**(a ) Original Cover Image**


**(b ) Stego-image with embedded data**

**(c ) Tampered Image**


**(d ) Extracted Image**

**Figure 3.3 Experimental result of a document image of a bank check attacked by erasing original image data**

(a ) Original Cover Image ; (b ) Stego-image with embedded data;
(c ) Tampered Image yielded by erasing original image data; (d ) Erroneous Extracted Image obtained with wrong key used for repairing

## 4. Performance Evaluation

Performance of proposed Secrete sharing based method for ensuring authenticity of gray scale document images is evaluated using various parameters like; ***Tampering Ratio, Detection ratio, Repair ratio, Mean Squared Error(MSE), Root Mean Squared Error(RMSE), Signal To Noise Ratio(SNR) and Peak Signal To Noise Ratio(PSNR).***

**4.1 Tampering Ratio =** ( The Number of Tampered Blocks ) / ( The Total Number of Blocks )

**4.2 Detection Ratio =** ( The Number of Detected Blocks ) / ( The Number of Tampered Blocks )

4.3 **Repair Ratio =** ( The Number of Repaired Blocks ) / ( The Number of Detected Blocks)

Table 4.1 shows Statistics of experimental results of proposed method using different 8 gray scale document images of paper and bank checks using three parameters. i.e. Tampering Ratio, Detection Ratio, Repair Ratio.

| Sr. No. | Input Image | Total No. of blocks | Total no. of Tampered Blocks (Tampering Ratio) | Total No. of Tampered Detected Bocks (Detection Ratio) | Total No. of Repaired Blocks (Repair Ratio) | Total No. of Unrepaired Blocks |
|---|---|---|---|---|---|---|
| 1 | Test 1 (as shown in figure 3.1) | 27400 | 1022 (3.72%) | 1022 (100%) | 852 (83.36%) | 170 |
| 2 | Test 2 | 30200 | 3559 (11.78%) | 3559 (100%) | 2966 (83.33%) | 593 |
| 3 | Test 3 (as shown in figure 3.2) | 30100 | 486 (1.61%) | 486 (100%) | 405 (83.33%) | 81 |
| 4 | Test 4 | 27900 | 941 (3.37%) | 941 (100%) | 784 (83.31%) | 157 |
| 5 | Test 5 | 50806 | 2516 (4.95%) | 2516 (100%) | 2097 (83.34%) | 419 |
| 6 | Test 6 (as shown in figure 3.3) (Wrong Key is used for Data repairing) | 50806 | 13117 (25.81%) | 13117 (100%) | 10931 (83.33%) | 2186 |
| 7 | Test 7 | 28700 | 0 | 0 | 0 | 0 |
| 8 | Test 8 | 55454 | 368 (.66%) | 368 (100%) | 3007 (83.42%) | 61 |

**Table 4.1 Shows Statistics of Experimental Results of proposed method using different 8 gray scale document images of paper and bank checks**

## 5. COMPARISION OF PERFORMANCE WITH OTHER METHODS

Table 5.1 shows comparative study of proposed Secret sharing based method for ensuring authenticity of gray scale images with earlier Image Authentication Methods

| Sr. No. | Author | Distortion in stego-image | Tampering localization capability | Repair capability | Reported authentication precision | Distribution of authenticated image parts | Manipulation of data embedding |
|---|---|---|---|---|---|---|---|
| 1 | Chih-HsuanTzeng and Wen-Hsiang Tsai [1] | Yes | Yes | No | 64 * 64 block | Entire image | Pixel replacement |
| 2 | H.Yang and A.C.Kot [2] | Yes | Yes | No | 33 * 33 block | Non blank part | Pixel flippability |
| 3 | H.Yang and A.C.Kot [3] | Yes | No | No | Macro block | Non blank part | Pixel flippability |
| 4 | M Wu and B. Liu [4] | Yes | No | No | Macro block | Non blank part | Pixel flippability |
| 5 | Proposed Method | No | Yes | Yes | 2* 3 block | Entire Image | Alpha channel pixel replacement |

**Table 5.1** Shows Comparative Study of Proposed Method with Earlier Image Authentication Methods

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude towards my guide, **Prof. Smita Deshmukh** and co-guide**, Dr. Deven Shah** for the help, guidance and encouragement they provided.

**CONCLUSION**

We have proposed a secure image authentication scheme for authentication of gray scale document images based on secret sharing method and chaotic logistic map. In this secure image authentication scheme security is provided by, secret sharing and encryption.

Using Shamir secret sharing method of shares creation both the generated authentication signal and the content of an image block are transformed into partial shares. Which are then distributed or embedded in an elegant manner into an alpha channel plane to create a PNG image. This image is encrypted by using chaotic logistic map and forms a stego image.

In the image authentication process, if it seen that the data is tampered then self-repairing is done in the content of the tampered block by reverse Shamir scheme. This method enhances the security by embedding the data in the alpha channel plane and encrypting the PNG image and hence provides two fold securities to gray scale document images.

**REFERENCES**

[1] Chih-HsuanTzeng and Wen-Hsiang Tsai, "A new approach to authentication of Binary image for multimedia communication with distortion reduction and security enhancement. IEEE communication letters VOL.7.NO.9, 2003.

[2] H. Yang and A. C. Kot, "Binary image authentication with tampering localization by embedding cryptographic signature and block identifier," IEEE Signal Processing Letters, vol. 13, Dec. 2006

[3] H. Yang and A. C. Kot, "Pattern based data hiding for binary image authentication by connectivity preserving," IEEE Trans. Multimedia, vol. 9, Apr. 2007

[4] M. Wu and B. Liu, "Data hiding in binary images for authentication and annotation," IEEE Trans. Multimedia, vol. 6, Aug. 2004

[5] Che- Wei Lee and Wen-Hsiang Tsai "A secret-sharing-based method for authentication of grayscale document images via the use of the png image with data repair capability" IEEE Trans. Image Processing., vol.21, no.1, january.2012.

[6] Niladri B. Puhan, Anthony T. S. Ho "Binary Document Image Watermarking for Secure Authentication Using Perceptual Modeling" IEEE International Symposium on Signal Processing and Information Technology 2005.

[7] A. Shamir, "How to share a secret," Commun. ACM, vol.22, no.11, pp.612-613, Nov, 1979.

[8] 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, june.2002.

[9] C Yu, X Zhang "Watermark embedding in binary images for authentication", IEEE Trans. Signal Processing, vol.01, no.07, pp.865-868, September. 2004