Secure Authentication using Image Processing and Visual Cryptography for Banking Applications

Chetana Hegde #[1], Manu S #[2], P Deepa Shenoy #[3], Venugopal K R #[4], L M Patnaik *[5]

# *Department of Computer Science and Engineering*
University Visvesvaraya College of Engineering, Bangalore University, Bangalore 560 001, India
[1] chetanahegde@yahoo.co.in
* *Defence Institute of Advanced Technology, Deemed University, Pune, India*

***Abstract*— Core banking is a set of services provided by a group of networked bank branches. Bank customers may access their funds and perform other simple transactions from any of the member branch offices. The major issue in core banking is the authenticity of the customer. Due to unavoidable hacking of the databases on the internet, it is always quite difficult to trust the information on the internet. To solve this problem of authentication, we are proposing an algorithm based on image processing and visual cryptography. This paper proposes a technique of processing the signature of a customer and then dividing it into shares. Total number of shares to be created is depending on the scheme chosen by the bank. When two shares are created, one is stored in the Bank database and the other is kept by the customer. The customer has to present the share during all of his transactions. This share is stacked with the first share to get the original signature. The Correlation method is used to take the decision on acceptance or rejection of the output and authenticate the customer.**

I. Introduction

Today, most applications are only as secure as their underlying system. Since the design and technology of middleware has improved steadily, their detection is a difficult problem. As a result, it is nearly impossible to be sure whether a computer that is connected to the internet can be considered trustworthy and secure or not. The question is how to handle applications that require a high level of security, such as core banking and internet banking.

In a core banking system, there is a chance of encountering forged signature for transaction. And in the net banking system, the password of customer may be hacked and misused. Thus security is still a challenge in these applications. Here, we propose a technique to secure the customer information and to prevent the possible forgery of signatures and password hacking.

The concept of image processing and an improved visual cryptography is used. Image processing is a technique of processing an input image and to get the output as either improved form of the same image and/or characteristics of the input image. Visual Cryptography (VC) is a method of encrypting a secret image into shares such that stacking a sufficient number of shares reveal the secret image.

Naor and Shamir [1] introduced a simple but perfectly secure way that allows secret sharing without any cryptographic computation, termed as Visual Cryptography Scheme (VCS). Basically, Visual Cryptography Scheme is an encryption method that uses combinatorial techniques to encode secret written materials. The idea is to convert the written material into an image and encode this image into n shadow images. The decoding requires only selecting some subset of these n images, making transparencies of them, and stacking them on top of each other.

The simplest Visual Cryptography Scheme is given by the following setup. A secret image consists of a collection of black and white pixels where each pixel is treated independently. To encode the secret image, we split the original image into n modified versions (referred as shares) such that each pixel in a share now subdivides into n black and white sub-pixels. To decode the image, a subset S of those n shares are picked and copied on separate transparancies. If S is a *qualified* subset, then stacking all these transparencies will allow visual recovery of the secret.

This paper is organized as follows: Section II deals with the related work and Section III presents the architecture and model. Section IV is about the problem definition. Section V presents the implementation of the proposed algorithm and the performance analysis. Section VI contains the conclusions.

II. RELATED WORK

A brief survey of the related work in the area of visual cryptography and its application in banking sector is presented in this section. Visual cryptography schemes were independently introduced by Shamir [2] and Blakley [3], and their original motivation was to safeguard cryptographic keys from loss. These schemes also have been widely employed in the construction of several types of cryptographic protocols [4] and consequently, they have many applications in different areas such as access control, opening a bank vault, opening a safety deposit box, or even launching of missiles. A segment-based visual cryptography suggested by Borchert [5] can be used only to encrypt the messages containing symbols, especially numbers like bank account number, amount etc.,

978-1-4244-2963-9/08/$25.00 © 2008 IEEE

the VCS proposed by Wei-Qi Yan et al., [6] can be applied only for printed text or image. A recursive VC method proposed by Monoth et al., [7] is computationally complex as the encoded shares are further encoded into number of sub-shares recursively. Similarly a technique proposed by Kim et al., [8] also suffers from computational complexity, though it avoids dithering of the pixels.

Most of the previous research work on VC focused on improving two parameters: pixel expansion and contrast [9],[10],[11]. In these cases all participants who hold shares are assumed to be honest, that is, they will not present false or fake shares during the phase of recovering the secret image. Thus, the image shown on the stacking of shares is considered as the real secrete image. But, this may not be true always. So, a cheating prevention methodologies are introduced by Yan et al., [12], Horng et al., [13] and Hu et al., [14]. But, it is observed in all these methodologies, there is no facility of authentication testing.

In this paper, we propose a VC method based on pixel matrices and a method for authentication.

III. ARCHITECTURE AND MODELING

In all banking applications, the customer has to create an account in one branch initially. The signature of the customer will be in the application form. We will consider this as our input. The flow of the algorithm is as follows. The signature of the customer is scanned from the application form. This image is pre-processed to thicken the light-shaded parts of the signature and to increase the intensity of the image. In the next stage, the pre-processed image is encrypted to get shares. The number of shares to be created is based on the scheme opted by the bank and the mode of operation of the account. The following schemes are developed:

- 2 out of 2 Scheme: This scheme can be adopted when the operational mode of the account is single. Here, two shares are created and both are necessary for decrypting the image. One of the shares is stored in the database of the bank and the other is kept with the customer.
- 2 out of 3 Scheme: It is useful for joint accounts. Here, three shares are created and any two are sufficient to reveal the image. Two shares are handed-over to two customers having joint account and the other is kept in bank database. Any one customer can transact with the bank by submitting his share.
- 3 out of 3 Scheme: This scheme is also useful for joint accounts. All three shares are required to get the output image in this case, and thus both the customers should be present for further transactions.
- Key-Share Scheme: It uses the signatures of both the customers in joint account system. From two images, four shares are created in the similar manner of 2 out of 2 scheme. Then by combining two shares, each from two images, a key-share is created. Overlaying key-share on each of the remaining shares will separately reveal two images. Thus, each customer can separately be authenticated and allowed to do transaction just by revealing his signature only.

In all the cases, the shares are printed and handed-over to customer, just like any credit/debit card. During further transaction, customer has to produce his share. This will be overlayed over the bank's share to decrypt the image. After decrypting the signature, the post-processing is done on the decrypted image. Finally, the original image and the post-processed image are compared to confirm the authenticity of the customer.

In a core banking system, the customer needs to present his share for every transaction. This share is overlayed on bank's share to get the signature and thus to authenticate the customer. ATM machines should allow scaning facility so that the customer may scan his share at ATM. Customer may be authenticated based on password and the output image obtained using the share produced by him.

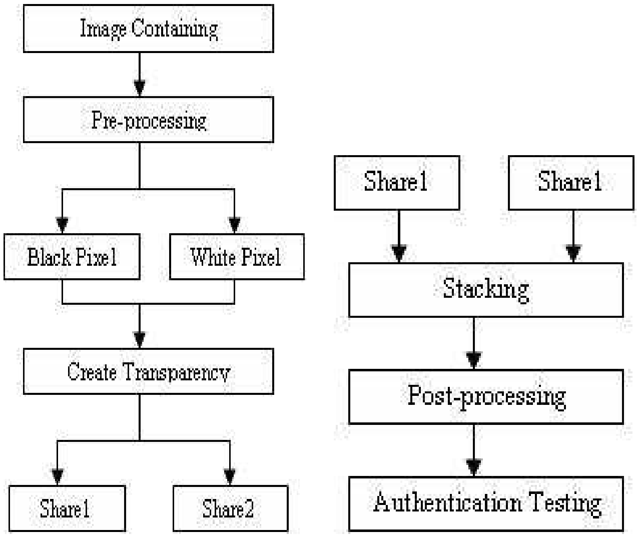
The model for creating shares is explained here. Let $\mathcal{P} = \{1, ..., n\}$ be a set of elements called *participants*, and let $2^{\mathcal{P}}$ denote the set of all subsets of $\mathcal{P}$. Let $\Gamma_{Qual} \subseteq 2^{\mathcal{P}}$ and $\Gamma_{Forb} \subseteq 2^{\mathcal{P}}$, where $\Gamma_{Qual} \cap \Gamma_{Forb} = \emptyset$. We refer to the members of Γ_{Qual} as *qualified sets* and the members of Γ_{Forb} as *forbidden sets*. The pair $(\Gamma_{Qual}, \Gamma_{Forb})$ is called as *access structure* of the scheme.

A participant $P \in \mathcal{P}$ is an *essential* participant if there exists a set $X \subseteq \mathcal{P}$ such that $X \cup P \in \Gamma_{Qual}$ but $X \notin \Gamma_{Qual}$. If a participant P is not essential, then we can construct a visual cryptography scheme giving him a share completely *white* or even nothing as his share. In fact, a non-essential participant does not need to participate *actively* in the reconstruction of the image, sicne the information he has is not needed by any set in $\mathcal{P}$ in order to recover the shared image. In any VCS having non-essential participants, these participants do not require any information in their shares. But, in our technique we consider that all participants are essential. In 2 out of 2 scheme, the participants are Bank and costomer. In all other schemes, the participants are Bank and two costomers. We have even developed the shemes like 3 out of 4, 4 out of 4, 3 out of 5, 4 out of 5 and so on. When there is a joint account with more than two customers, one can go for such extended schemes.

The architecture for the process of share creation for 2 out of 2 scheme is shown in Fig. 1(a) and that for testing authentication is shown in Fig. 1(b). The basic assumption for this algorithm is that the scanned image is a gray-scale image.

A. Pre-Processing

The scanned image is initially converted into grey-scale image. Some of the test images are shown in Fig. 2. Then the image is thresholded. The threshold value is chosen automatically. Every image say, *f(x, y)* is composed of light



(a) Share Creation

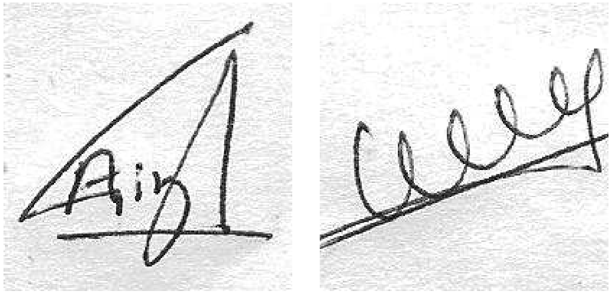
(b) Authentication

Fig. 1. Architecture for 2 out of 2 Scheme

objects on a dark background, in such a way that object and background pixels have intensity levels grouped into two dominant modes. One obvious way to extract the objects from the background is to select a threshold *T* that separates these modes. Then any point (*x, y*) for which *f(x, y)* $\geq$ *T* is called an object point; otherwise the point is called a background point. In other words, the threshold image *g(x,y)* is defined as

$$g\left(x,y\right) = \begin{cases} 1 & \text{if} \quad f\left(x,y\right) \geq T \\ 0 & \text{if} \quad f\left(x,y\right) < T \end{cases}$$

Pixels labelled 1 correspond to objects, whereas pixels labelled 0 correspond to the background. This approach is global thresholding, when *T* is constant. Global thresholding is useful only when the background illumination of the image is even. Otherwise, the image must be pre-processed to compensate for the illumination problems and then global threshold should be applied. This method is local thresholding. This process is given by



(a) Signature1

(b) Signature2

Fig. 2. Original Images

$$g\left(x,y\right) = \begin{cases} 1 & \text{if} \quad f\left(x,y\right) \geq T\left(x,y\right) \\ 0 & \text{if} \quad f\left(x,y\right) < T\left(x,y\right) \end{cases}$$

Where, $T\left(x,y\right)$ is a locally varying threshold function. The formula to compute it is given by

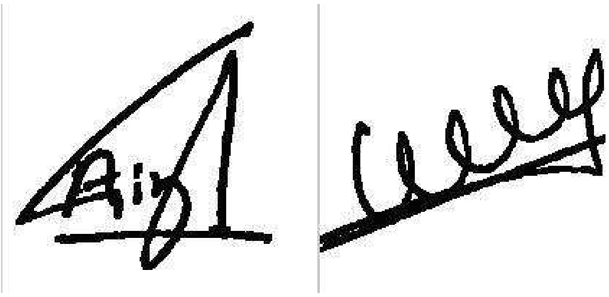
$$T\left(x,y\right) = f_0\left(x,y\right) + T_0$$

The image $f_0\left(x,y\right)$ is the morphological opening of *f* and the constant T_0 is the result of global threshold applied on f_0.

Morphological erosion is applied on the thresholded image. Erosion is to shrink the image using a structuring element. This can be given as

$$A \ominus B = \{z \mid \left(B\right)_z \cap A^c \neq \emptyset\}$$

In other words, erosion of *A* by *B* is the set of all structuring element origin locations where the translated B has no overlap with the background of *A*. The images after pre-processing are shown in Fig. 3.



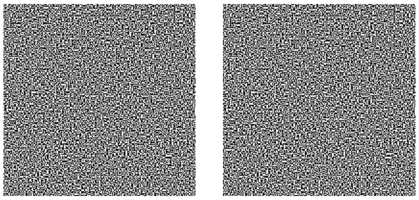
(a) Signature1

(b) Signature2

Fig. 3. Images after Pre-processing

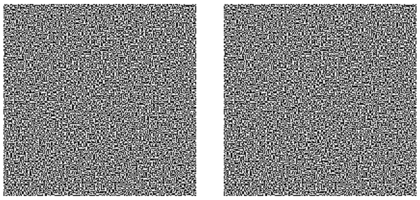
B. Creation of Shares

The basic assumption here is that the image is a collection of black and white pixels and each pixel is handled separately. Each original pixel appears in *n* modified versions, one for each share. Each share is a collection of *m* black and white sub-pixels that are printed in close proximity to each other so that the human visual system averages their individual black/white contributions. The resulting structure can be described by an $n \times m$ Boolean matrix $S = [s_{ij}]$ where s_{ij} =1 iff the j^{th} sub-pixel in the i^{th} share is black. Fig. 4 to Fig. 7 shows the shares of the image Signature1, obtained in various schemes.



(a) Share1 (b) Share2

Fig. 4. Two shares obtained for the image Signature1 in 2 out of 2 Scheme



(a) Share1 (b) Share2 (c) Share3

Fig. 5. Three shares obtained for the image Signature1 in 2 out of 3 Scheme

C. Stacking

It is a procedure of getting original image by stacking the transparencies. When transparencies $i_1, i_2, ..., i_r$ are stacked together in a way which properly aligns the sub-pixels, one can see a combined share whose black sub-pixels are represented by the Boolean *OR* of rows $i_1, i_2, ..., i_r$ in *S*. The grey level of this combined share is proportional to the Hamming weight $H(V)$ of the *OR*ed *m*-vector *V*. This grey level is interpreted by the visual system of the users as black if $H(V) \geq d$ and as white if $H(V) < d - \alpha m$ for some fixed threshold $1 \leq d \leq m$ and relative difference $\alpha > 0$. The images after stacking the shares are shown in Fig. 8 and Fig. 9.

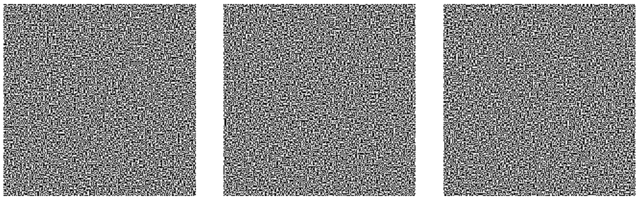
D. Post-processing

The stacked shares results in an image where the black pixels will yield the required information. But other pixels are randomly distributed. To avoid this noise, the technique of post-processing is applied on this image. Morphological closing is applied initially. Closing of an image *A* by a structuring element *B* is a dilation followed by erosion, which is given as

$$A \bullet B = (A \oplus B) \ominus B$$

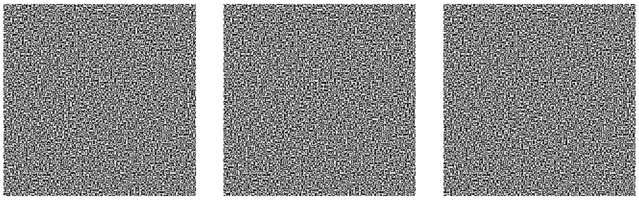
Then order-statistics filters are used, whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter. The response of the filter at any point is then determined by the ranking result. The current algorithm uses Median filter, the best-known order-statistics filter, which is given as

$$\hat{f}(x, y) = median_{(s,t) \in S_{xy}} \{g(s, t)\}$$



(a) Share1 (b) Share2 (c) Share3

Fig. 6. Three shares obtained for the image Signature1 in 3 out of 3 Scheme



(a) Key Share (b) Share1 (c) Share2

Fig. 7. Three shares obtained for the image Signature1 in Key-Share Scheme

The image after post-processing is as shown in Fig. 10.

E. Authentication Testing

After obtaining the signature, it is tested for the authenticity. If the shares of different signatures are stacked, an absurd image is obtained as shown in Fig. 11. A possible attempt made to cheat the bank may thus be overruled. There is a possibility of producing a share which will result in some signature format, but not the actual signature [14]. Such an attempt is overruled by comparing the decrypted signature with the original signature. This algorithm uses the correlation technique for checking the authenticity.

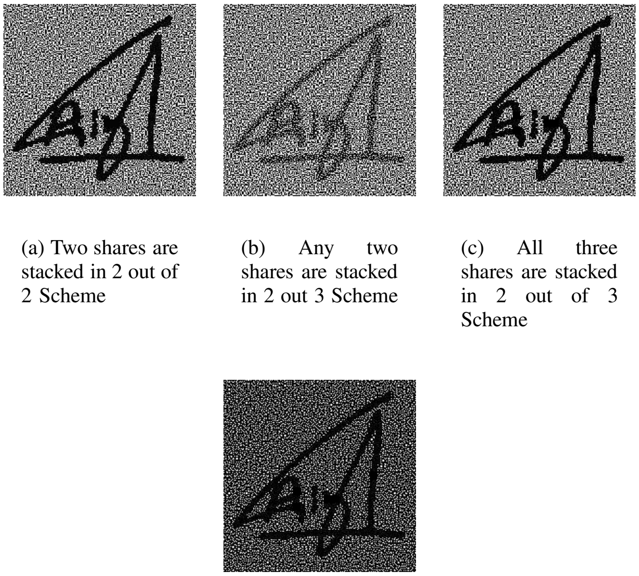
Correlation is a method of identifying the degree of relationship between two sets of values. Karl Pearsons correlation coefficient reveals the dependency or independency between the variables. If *X* and *Y* are two arrays, then the Karl Pearsons correlation coefficient between *X* and *Y* is computed using the formula

$$\rho_{XY} = \frac{E\left((X - \mu_X)(Y - \mu_Y)\right)}{\sigma_X \sigma_Y}$$

By simplifying the above formula, we will get

$$\rho_{XY} = \frac{E(XY) - \mu_X \mu_Y}{\sigma_X \sigma_Y}$$

Here, *E* is the *expected value* operator, μ_X, μ_Y and σ_X, σ_Y are means and standard deviations of *X* and *Y* respectively. The value of correlation coefficient ρ_{XY} may range from



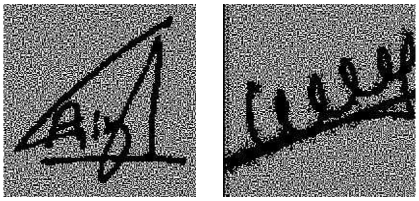
(a) Two shares are stacked in 2 out of 2 Scheme

(b) Any two shares are stacked in 2 out 3 Scheme

(c) All three shares are stacked in 2 out of 3 Scheme

(d) Three shares are stacked in 3 out 3 Scheme

Fig. 8. Stacked Shares for Signature1



(a) Key Share is stacked with Share1

(b) Key Share is stacked with Share2

Fig. 9. Stacked Shares for Signature1 and Signature2 in Key-Share Scheme

-1 to +1. If the value of correlation coefficient is -1, the variables X and Y are inversely related. If the value is 0, then the variables are independent and if the value is 1, then the variables are completely (or positively or directly) related. Thus, the high degree of positive correlation indicates that the values of variables are very much close to each other. So, if the correlation coefficient between the original image and the output image is nearer to +1, authenticity may be granted. If the correlation coefficient is nearer to zero, one can decide that the share produced by customer is fake and can be rejected.

IV. ALGORITHM

A. Problem Definition

Given an image in the format of GIF or JPEG (200×200), the objectives are:

(1) To encrypt it and to get either two or three shares based on the scheme opted.

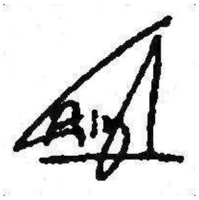


Fig. 10. The image Signature1 after Post-processing

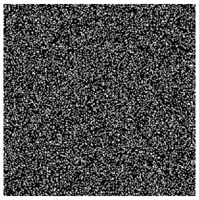


Fig. 11. Share1 of Signature1 and Share2 of Signature2 (2 out of 2 Scheme)are stacked

(2) To stack the shares to get the image and

(3) To improve the clarity of the resulting image and then comparing it with the original image.

The basic assumption made here is that signature images are grey scale images.

B. Algorithm

Four major functions are involved in the proposed technique. The first function is to pre-process the given image to remove the possible noise and to get the clarity. The second function is to encrypt the image based on white and black pixels and to get two transparencies. The third function is the overlay/stack the available shares to get the image containing signature. The last function is to compare the original image with the resulting image using correlation technique. The aim of the algorithm is to design an efficient technique for checking authenticity of the customer in core-banking and internet banking applications. The algorithms/pseudo codes for various steps involved are shown here. Given an image containing signature, the initial problem is to convert it into a grey-scale image. Then it is processed to remove noise and to increase intensity. In the next step, the foils must be created for white pixel and black pixel. The black pixel, denoted by 1, is an information pixel and the white pixel, denoted by 0 represents background. The initial Boolean matrices for white pixel, S_0 and for black pixel, S_1 for two shares in 2 out of 2 Scheme are given here.

$$S_0 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

$$S_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

The methodology for creation of shares is explained here. Two rows of matrices denote the number of shares to be created. If a particular pixel is white in the original image, then two rows of the matrix S_0 are put into two shares, one for each. If the pixel is black in original image, then the rows of S_1 are used. Thus, a single pixel in original image takes four positions in the shares. So, all the shares will be four times the original image in size.

Decryption is achieved by stacking the shares. In case of black pixel, overlaying two rows of S_1 results in four black bits, and reveals the information. Where as for the white pixel, stacking the two rows of S_0 results in two black and two white bits, and thus introduces noise. To overcome this noise, we have to post-process the output image.

The boolean matrices for three shares in 2 out 3 Scheme are:

$$S_0 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$S_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

As 2 out of 3 scheme needs three shares to be created, the Boolean matrices have 3 rows, one for each share. One can note that, stacking of any two rows of the matrix S_1 results all four black bits and thus reveal information. Stacking of any two rows in S_0 will always result in mixture of white and black bits.

The boolean matrices for three shares in 3 out 3 Scheme are:

$$S_0 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$S_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

The 3 out of 3 scheme also needs three shares to be created. Here, the information bits, that is, black bits can be retrieved only after stacking all three rows in S_1. Overlaying any two shares will not reveal any information about the original image.

To have Key-Share scheme, initially, two signatures must be encrypted using 2 out of 2 scheme. This results in four shares, say, C1, C2, C3 and C4. Now, C1 and C3 are combined to get a key share. Overlaying this key share on C2 results in first signature and on C4 results in second signature image.

One of such shares is stored in the database of the bank and the other is printed and given to the customer. In case of joint account, one share will remain with the bank and other two shares are distributed to customers. During further transactions, the share submitted by the customer is stacked over the share in the bank. To get the image, again the Boolean matrices are used. The produced image is post-processed to remove the noise and then compared with the original image.

TABLE I

Algorithm : Increasing Intensity of the Image through Pre-processing

```
Step 1:Select an initial estimate for the threshold $T$.
Step 2:Segment the image using $T$. This will produce two
        groups of pixels: $G_1$ consisting of all pixels with
        grey level values $> T$ and $G_2$ consisting of pixels
        with values $\leq T$.
Step 3: Compute the average grey level values $\mu_1$ and $\mu_2$
        for the pixels in regions $G_1$ and $G_2$.
Step 4: Compute a new threshold value:

                    $T = \frac{1}{2}(\mu_1 + \mu_2)$

Step 5: Repeat steps 2 to 4 until the difference in $T$
        in successive iterations is smaller than a
        predefined parameter $T_0$
```

TABLE II

Algorithm : Creation of Shares for 2 out of 2 Scheme

```
Step 1: Create two matrices $S_0$ and $S_1$ for white and black
        pixels.
Step 2: Initialize two variables $WHITEPIXEL$ and
       $BLACKPIXEL$.
Step 3: for $i = 1$ to $rows$
          for $j = 1$ to $columns$
            for $k = 0$ to 3
              if $Img(i,j)$==$WHITEPIXEL$
                set $Share1(i,j+k)$=$WHITEPIXEL$
                set $Share2(i,j+k)$=$WHITEPIXEL$
              else
                set $Share1(i,j+k)$=$BLACKPIXEL$
                set $Share2(i,j+k)$=$BLACKPIXEL$
              end if
            end for
          end for
        end for
```

Table I gives the algorithm to improve the intensity of the image. Initially, an arbitrary estimate for the threshold T is selected. Then the image is segmented using T in such a way that two groups of pixels will be produced. One of these groups consists of all pixels with grey level values greater than T and the other group contains the pixels with grey level values less than or equal to T. Now the average grey level values are computed and then new threshold value is calculated. This procedure of computing new threshold is continued till it becomes smaller than a predefined parameter T_0.

TABLE III

Algorithm : Stacking the Shares for 2 out of 2 Scheme

```
$Stack()$
begin
   for $i = 1$ to $rows$
      for $j = 1$ to $columns$
         for $k = 0$ to 3
            if $Share1\,(i, j + k)$==$BLACKPIXEL$
               set $OutImg\,(i, j + k)$=$WHITEPIXEL$
            else
               set $OutImg\,(i, j + k)$=$BLACKPIXEL$
            end if
         end for
      end for
   end for
end
```

TABLE IV

Algorithm : Calculate Correlation Coefficient

```
$Corr - Coeff()$
//Input: The original image $X$ and the resulting image $Y$.
//Output: The Correlation Coefficient between $X$ and $Y$.

begin
   Intialize $SumX = 0$, $SumY = 0$, $SumSqX = 0$,
   $SumSqY = 0$and $SumXY = 0$
   for $i = 1$ to $rows$
      for $j = 1$ to $columns$
         set $SumX = SumX + X\,(i, j)$
         set $SumY = SumY + Y\,(i, j)$
         set $SumSqX = SumSqX + X\,(i, j)^2$
         set $SumSqY = SumSqY + Y\,(i, j)^2$
         set $SumXY = SumXY + X\,(i, j) * Y\,(i, j)$
      end for
   end for

   $AvgX = SumX/(rows * cols)$
   $AvgY = SumY/(rows * cols)$
   $EXY = SumXY/(rows * cols)$
   $StdX = sqrt(SumSqX/(row * cols)AvgX^2)$
   $StdY = sqrt(SumSqY/(row * cols)AvgY^2)$
   $Corr = (EXYAvgX * AvgY)/(StdX * StdY)$

end
```

Algorithm in Table II is to create shares for a given signature image. Initial matrices S_0 and S_1 are created for white and black pixels. Depending on the value of any pixel in the i^{th} row of image, four pixels in that row are set as per the matrices for both the shares. Now, these two shares cannot individually be used to get the secret image.

Algorithm $Stack()$ in Table III is to overlap two shares to get the secret image. Table IV reveals the calculation of Karl Pearson's Correlation Coefficient between the original image and the resulting secret image.

V. IMPLEMENTATION AND PERFORMANCE ANALYSIS

The implementation of pre-processing, post-processing and authentication testing using correlation method is done using Matlab 7. Creation of shares and stacking of shares are implemented in Java (Jdk1.5). It is observed through simulation that both original image and the output image for test signatures are related with very high degree of correlation. The actual values of correlation coefficients for different signature images are given in Table V. The graph showing the degree of correlation between original image and the output image for Signature1 is shown in Fig. 12. We observe that, the correlation is in positive direction with very less scatteredness among values.

When the shares of different signatures are stacked, then there will be zero degree of correlation between original and output images. The correlation coefficient between the image Signature1 and an absurd image of Fig. 11 is $-8.2682e-5 \approx 0$. The graph in Fig. 13 depicts the scatteredness and thus proves the independence.

TABLE V

Test Images	Correlation Coefficient
Signature1	0.9089
Signature2	0.9123
Signature3	0.8540
Signature4	0.9782

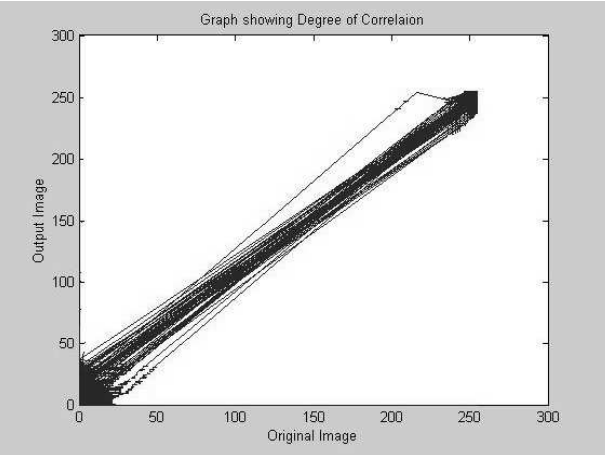


Fig. 12. Graph showing High Degree of Positive Correlation for Signature1

VI. CONCLUSION

In this paper, we propose an efficient way to improve security in core banking and net banking applications. Initially, the signature of the customer is pre-processed to improve the intensity and then scanned. The image is divided into two shares. One of the shares is stored in bank database and the other is handed-over to the customer. During further transactions, he is supposed to submit his share. The

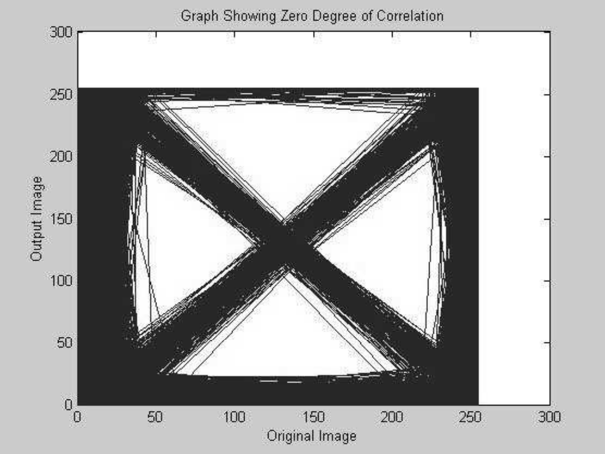


Fig. 13. Graph showing Zero Degree of Correlation

submitted share is stacked on the other share of the bank. The decrypted image is tested for authentication using correlation technique. The experimental results obtained indicate that the genuine shares submitted indicates high degree of positive correlation and thus support authenticity, and the fake shares produce zero degree of correlation. In the future work, we can consider the colour images and try to improve the quality of decrypted image. In this paper, we have considered signature of the customer as the input. But, any other image accepted by both bank and customer can fairly be replaced as input.

References

[1] M. Naor and A. Shamir, ”Visual Cryptography,” *Advances in Cryptography -EUROCRYPT’94*, Lecture Notes in Computer Science 950, 1995, pp. 1-12.

[2] A. Shamir, ”How to Share a Secret,” *Communication ACM*, vol. 22, 1979, pp. 612-613.

[3] G. R. Blakley, ”Safeguarding Cryptographic Keys,” *Proceedings of AFIPS Conference*, vol. 48, 1970, pp. 313-317.

[4] A. Menezes, P. Van Oorschot and S. Vanstone, ”Handbook of Applied Cryptography,” CRC Press, Boca Raton, FL, 1997.

[5] B. Borchert, ”Segment Based Visual Cryptography,” WSI Press, Germany, 2007.

[6] W-Q Yan, D. Jin and M. S. Kanakanahalli, ”Visual Cryptography for Print and Scan Applications,” *IEEE Transactions*, ISCAS-2004, pp. 572-575.

[7] T. Monoth and A. P. Babu, ”Recursive Visual Cryptography Using Random Basis Column Pixel Expansion,” in *Proceedings of IEEE-International Conference on Information Technology*, 2007, pp. 41-43.

[8] H. J. Kim, V. Sachnev, S. J. Choi and S. Xiang, ”An Innocuous Visual Cryptography Scheme,” in *Proceedings of IEEE-*8th *International Workshop on Image Analysis for Multimedia Interactive Services*, 2007.

[9] C. Blundo and A. De Santis, ”On the contrast in Visual Cryptography Schemes,” in *Journal on Cryptography*, vol. 12, 1999, pp. 261-289.

[10] P. A. Eisen and D. R. Stinson, ”Threshold Visual Cryptography with specified Whiteness Levels of Reconstructed Pixels,” *Designs, Codes, Cryptography*, vol. 25, no. 1, 2002, pp. 15-61.

[11] E. R. Verheul and H. C. A. Van Tilborg, ”Constructions and Properties of k out of n Visual Secret Sharing Schemes,” *Designs, Codes, Cryptography*, vol. 11, no. 2, 1997, pp. 179-196.

[12] H. Yan, Z. Gan and K. Chen, ”A Cheater Detectable Visual Cryptography Scheme,” *Journal of Shanghai Jiaotong University*, vol. 38, no. 1, 2004.

[13] G. B. Horng, T. G. Chen and D. S. Tsai, ”Cheating in Visual Vryptography,” *Designs, Codes, Cryptography*, vol. 38, no. 2, 2006, pp. 219-236.

[14] C. M. Hu and W. G. Tzeng, ”Cheating Prevention in Visual Cryptography,” *IEEE Transaction on Image Processing*, vol. 16, no. 1, Jan-2007, pp. 36-45.