LOSSLESS RECONSTRUCTION OF SECRET IMAGE USING THRESHOLD SECRET SHARING AND TRANSFORMATION

P. Devaki[1] , Dr. G. Raghavendra Rao[2]

[1]Department of Information Science & Engineering, NIE, Mysore
p_devaki1@yahoo.com
[2]Department of Computer Science & Engineering, NIE, Mysore
grrao57@gmail.com

ABSTRACT

This paper is proposed to provide confidentiality of the secret image which can be used by multiple users or to store on multiple servers. A secret sharing is a technique to protect the secret information which will be used by multiple users. The threshold secret sharing is more efficient as it is possible to reconstruct the secret with the threshold number of shares. Along with Shamir's secret sharing method we propose to use the radon transformation before dividing the image in to shares. This transformation is used so that the shares will not have the original pixel intensity. The run length code is used to compress the image after the transformation. Then apply secret sharing technique. The reconstruction of the image results in original image by applying the operations in the reverse order.

KEYWORDS

Threshold secret sharing, Confidentiality, Transformation, Compression.

1. INTRODUCTION

Handling secret information in a public network like internet is not easy. It is not safe to store the secret information on a system which can be accessed by multiple users. Even if the secret information stored in a system which is protected with a password also, there may be a single point of failure. It is not completely protected because multiple users will have the password. It may be misused or compromised by one or multiple users with out the knowledge of other users. This compromises the confidentiality of the information. If the secret information needs to be transmitted over a network, it can not be sent as it is over a network. There are 2 reasons for this restriction. One reason is the bandwidth required to send the information. Second reason is the issue about maintaining the confidentiality of the secret information during the transmission over the network.

To overcome these problems of handling secret information, an efficient method proposed by Shamir [4] and Blakley [5] can be used. This method is nothing but instead of allowing all the authorized users to have the accessibility to the secret information or transmitting the whole information as a single unit of data, divide the information in to number of blocks.

Each block is called as a share. The number of shares depends on the number of authorized users of that information. Each user will get a share of the information instead of the whole information. This ensures that even if a user misuses or compromises his/her share with the unauthorized user, it's of no use for the unauthorized user.

DOI : 10.5121/ijnsa.2012.4307

With a single share, the unauthorized user will not be able to guess the information. When the information is required by any of the users, that user can request and collect the shares from the other users and reconstruct the information.

This method is referred to as secret sharing. At no point of time any authorized user will not get the information about the length of the information and also the shares belonging to other users. In this conventional secret sharing method, all the shares are required to reconstruct the information. Even if one share is not available during the reconstruction process, it will not be possible to reconstruct the original information.

Threshold secret sharing [4] [5] [6] is an extension of the conventional secret sharing method. Threshold secret is nothing but, it is not necessary to collect all the shares to reconstruct the information. It is sufficient to collect the threshold number of shares to reconstruct the information. (k, n) indicates k out of n number of shares is sufficient to reconstruct the information. Here k is the threshold value, and n is the total number of shares of the information. With even k-1 number of shares it is not possible to reconstruct the information. It is flexible compared to the conventional secret sharing method, at the same time it is secured also. The information can be text, audio, video or image.

While reconstructing the secret information, the authorized users will come to know about the user who has requested for the secret shares. During this time the users can verify whether the requester is an authorized user or not.

In some applications which deal with the images, the reconstruction of the secret image can tolerate some loss. Even with the loss of some data also it is possible to recognize the image. Only the image needs to be visually recognizable. This is referred to as visual cryptography. Many researchers have worked in this area [9][10]. Here the quality of the reconstructed image will be degraded due to either bits transformed in to frequency domain or division of pixels. But in some applications it is necessary to reconstruct the image in such a way that there should not be any loss of data. Examples: medical, military, business documents, images taken by the satellite etc for which any loss of data is undesirable. Many researchers have been working towards this in providing security to the secret images [11][12].

For larger amount of information, it is necessary to compress the data before performing the sharing operation and distributing the shares. Compression must be lossless. Compression also provides a first level of confidentiality to the information. Since, it is not possible to recognize the compressed data. Then sharing can be made. This provides second level of confidentiality to the information. Before compressing the Image it is necessary to transform the image.

The transformations can be frequency or spatial based. There are various transformation methods available. By performing inverse transformation some methods result in the original image reconstruction, where as some result in some loss in the image but visually it can be recognized. The loss may be due to quantization error. But for applications like medical and military, it is necessary to reconstruct the original image rather than the image with loss of data.

In this paper we propose to use the threshold secret sharing based on Shamir. Radon transformation is used with run length coding to compress the image. The rest of the paper has been organized as follows. Concept of secret sharing is explained Section 2 , radon transformation is explained in section 3, compression using run length coding in section 4, proposed work in section 5, conclusion in section 6.

2. Threshold secret sharing

Shamir's threshold secret sharing[4] is based on the polynomial interpolation. He uses Lagrange's interpolation for reconstruction of the key. When it is necessary to distribute the secret information among multiple users, this secret sharing method can be used to generate the shares using the polynomial of the order m-1 for (m,n) secret sharing. m is the threshold value and n is the total number of shares to be done based on the number of authorized users.

The polynomial is

$f(x) = S + Cx + Cx^2 + \ldots\ldots.Cx^{m-1}$

Where S is the secret , to be shared among n users. C1, C2, C3 are the coefficients whose values are randomly selected. Using this polynomial and selecting unique values for x, the dealer divides the secret in to number of shares.
f(x1) = y1
f(x2) = y2
….
F(xn) = yn

Where y1, y2,…… yn are the shares generated. Now the shares along with the x values will be distributed to the authorized users. The dealer is not responsible for reconstruction of the key. So each user i will get a pair of values xi and yi.

When it is necessary for a user to reconstruct the information, he requests other users to send their shares. After receiving at least m number of shares the user can reconstruct the information. The significance of this secret sharing method is, no single share will reveal any information about the secret information, and with less than m number of shares also it is not possible to reconstruct the secret information. Only if a user gets minimum m number of shares from the authorized users, he can reconstruct the key. This will ensure that even if an attacker gets a share or few shares which is less than m, the attacker will not be able to reconstruct the secret information. This gives protection against any attacker getting the secret, and also protects against any authorized user miss using the secret.

The reconstruction of the secret is performed by using the Lagrange's interpolation. This is used to solve for the coefficients especially for S.

The interpolation formula is:

$$f(x) = \frac{(x-x1)(x-x2)\,y0}{(x0-x1)(x0-x2)} + \frac{(x-x0)(x-x2)y1}{(x1-x0)(x1-x2)} + \frac{(x-x0)(x-x1)\,y2}{(x2-x0)(x2-x1)} + \ldots\ldots\ldots\ldots\ldots$$

Solving for the above will result in S.

This is a perfect secret sharing, because the secret S is reconstructed only with m or more than m shares. The secret S can not be constructed with less than m number of shares.

As a simple example we have considered (3,5) where 5 is the number of authorized users, and 3 is the threshold. The given secret is 222.

The polynomial is
$F(x) = 222 + 3x + 2x^2$
We can select 5 values for x and calculate F(x).
Y1= F(x1) = 227
Y2 = F(x2) = 236
Y3 = F(x3) = 249
Y4 = f(x4) = 266
Y5 = F(x5) = 287

After obtaining the shares y1 to y5, the shares along with the values of x for each user can be given to the users as follows.

U1 = (x1, F(x1)) = (1, 227)
U2 = (x2, F(x2)) = (2, 236)
U3 = (x3, F(x3)) = (3, 249)
U4 = (x4, F(x4)) = (4,266)
U5 = (x5, F(x5)) = (5, 287)
When the 5 users get their respective shares, they can store them.

3. Radon transformation

The Radon transform is the projection of the image intensity along a radial line oriented at a specific angle. The radon function computes projections of an image matrix along specified directions.

R = radon(I, theta) returns the Radon transform R of the intensity image I for the angle theta degrees. If theta is a scalar, R is a column vector containing the Radon transform for theta degrees. If theta is a vector, R is a matrix in which each column is the Radon transform for one of the angles in theta.

[R,xp] = radon(I,theta) returns a vector xp containing the radial coordinates corresponding to each row of R.

The radial coordinates returned in xp are the values along the x'-axis, which is oriented at theta degrees counterclockwise from the x-axis. The origin of both axes is the center pixel of the image, which is defined as floor((size(I)+1)/2).

A projection of a two-dimensional function f(x,y) is a set of line integrals. The radon function computes the line integrals from multiple sources along parallel paths, or beams, in a certain direction. The beams are spaced 1 pixel unit apart. To represent an image, the radon function takes multiple, parallel-beam projections of the image from different angles by rotating the source around the center of the image. The following figure shows a single projection at a specified rotation angle.

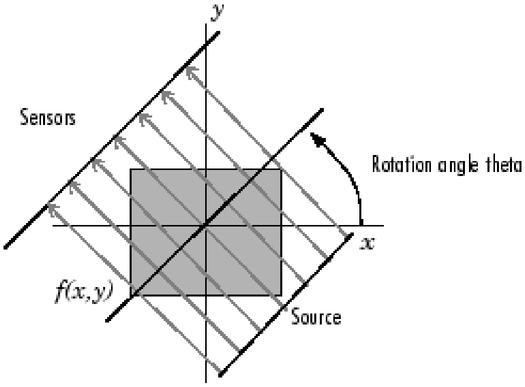

Fig -1

Projections can be computed along any angle [[THETA]]. In general, the Radon transform of *f(x,y)* is the line integral of *f* (x,y) parallel to the *y´*-axis

$$R_\theta(x') = \int_{-\infty}^{\infty} f\left(x'\cos\theta - y'\sin\theta,\ x'\sin\theta + y'\cos\theta\right) dy'$$

where

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

The following figure illustrates the geometry of the Radon transform.

Geometry of the Radon Transform

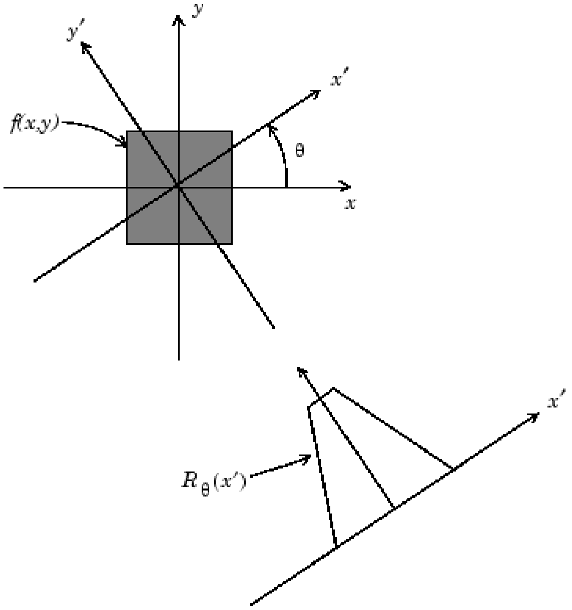

Fig – 2

The radial coordinates returned in xp are the values along the x'-axis, which is oriented at theta degrees counterclockwise from the x-axis. One important feature of this transformation is on retransformation it results in the original image.

Radon retransformation:
IR = iradon(R,theta);
iradon reconstructs an image from parallel-beam projections. In parallel-beam geometry, each projection is formed by combining a set of line integrals through an image at a specific angle.
I = iradon(R, theta) reconstructs the image I from projection data in the two-dimensional array R. The columns of R are parallel beam projection data. iradon assumes that the center of rotation is the center point of the projections, which is defined as ceil(size(R,1)/2).

theta describes the angles (in degrees) at which the projections were taken. It can be either a vector containing the angles or a scalar specifying D_theta, the incremental angle between projections. If theta is a vector, it must contain angles with equal spacing between them. If theta is a scalar specifying D_theta, the projections were taken at angles theta = m*D_theta, where m = 0,1,2,...,size(R,2)-1. If the input is the empty matrix ([]), D_theta defaults to 180/size(R,2).
iradon uses the filtered back-projection algorithm to perform the inverse Radon transform. The filter is designed directly in the frequency domain and then multiplied by the FFT of the projections. The projections are zero-padded to a power of 2 before filtering to prevent spatial domain aliasing and to speed up the FFT.
I = iradon(P, theta, *interp*, *filter*, frequency_scaling, output_size) specifies parameters to use in the inverse Radon transform. You can specify

any combination of the last four arguments. `iradon` uses default values for any of these arguments that you omit.
`interp` specifies the type of interpolation to use in the back projection. The available options are listed in order of increasing accuracy and computational complexity.

4. Runlength Encoding

Run-length encoding is a data compression algorithm that is supported by most bitmap file formats, such as Tiff, BMP, and PCX. RLE is suited for compressing any type of data regardless of its information content, but the content of the data will affect the compression ratio achieved by RLE. Although most RLE algorithms cannot achieve the high compression ratios of the more advanced compression methods, RLE is both easy to implement and quick to execute, making it a good alternative to either using a complex compression algorithm or leaving your image data uncompressed.

RLE works by reducing the physical size of a repeating string of characters. This repeating string, called a *run*, is typically encoded into two bytes. The first byte represents the number of characters in the run and is called the *run count*. In practice, an encoded run may contain 1 to 128 or 256 characters; the run count usually contains as the number of characters minus one (a value in the range of 0 to 127 or 255). The second byte is the value of the character in the run, which is in the range of 0 to 255, and is called the *run value*.

Ex:Uncompressed, a character run of 15 X characters would normally require 15 bytes to store

XXXXXXXXXXXXXXX

The same string after RLE encoding would require only two bytes

`15X`

The 15X code generated to represent the character string is called an *RLE packet*. Here, the first byte, 15, is the run count and contains the number of repetitions. The second byte, A, is the run value and contains the actual repeated value in the run.

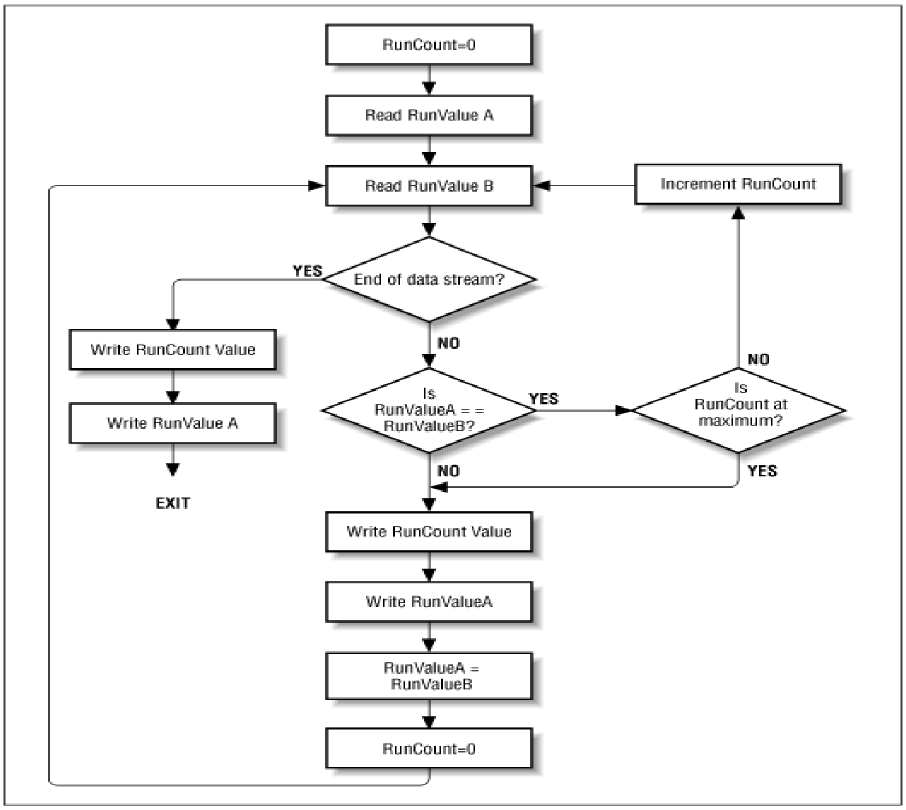

Fig - 3

There can be bit level, byte level and pixel level RLE.

5. Proposed work

If the image is transformed to frequency domain as many researchers have done, then quantization is required and also processing time is required for the conversion. Due to quantization the retransformation may not result in the original image. So we are using spatial domain transformation.

The given gray image is transformed using radon transform. Radon is a spatial domain transform, where the pixel's intensity will be oriented in a particular angle. Since the transformed image is different from the original image the confidentiality will be maintained.

This gives the first level of confidentiality to the secret image. Then the byte runlength code is applied to get the compressed transformed image. This gives second level of confidentiality. Then apply Shamir's secret sharing on this encoded data to get n number of shares. This gives third level of confidentiality.

The shares will be a sequence of numeric values. These values are represented in Braille form. The Braille images are sent to the authenticated users. This gives the fourth level of confidentiality. Here the purpose of using the Braille is to avoid using the cover images to hide the shares.

When it is necessary to obtain the image, atleast m shares which are in the Braille form is collected and transform Braille to normal sequence of numeric values. Then apply the Lagrange's interpolation to obtain the integer value of the compressed image. Then apply the run length decoding to obtain the transformed image. After these apply the radon retransform to obtain the original image.

Distribution of the secret image:

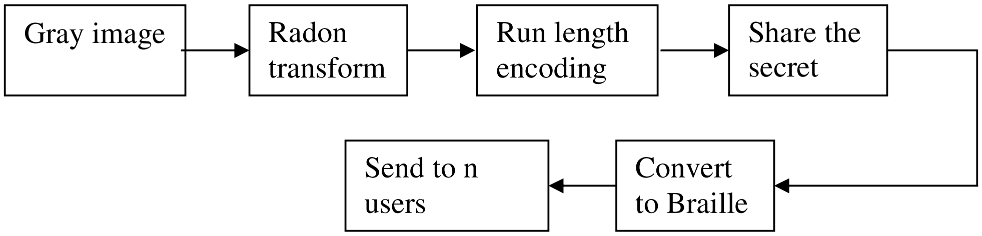

Fig - 4

Reconstruction of the image:

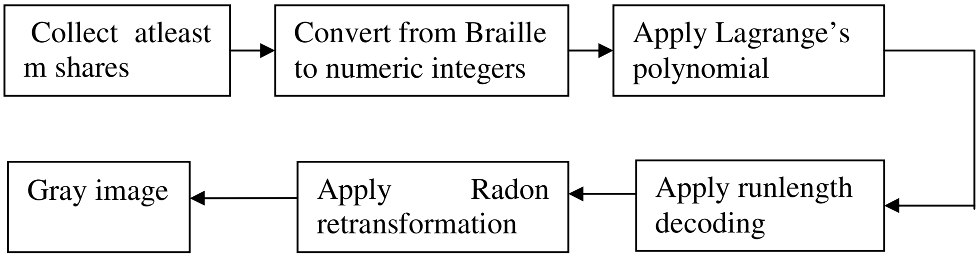
Collect atleast m shares
Convert from Braille to numeric integers
Apply Lagrange's polynomial
Apply runlength decoding
Apply Radon retransformation
Gray image

Fig - 5

6. Conclusions

The proposed work is fault tolerant, as even if few shares are corrupted also it is possible to reconstruct the image with m shares. Unlike the visual cryptography, this method allows to reconstruct the original image. The size of the shares is not the size of the original image, as the image is compressed. This reduces the bandwidth required to transmit the shares on the network and also the space required to store the shares. It is highly secured, as the image is transformed, compressed and then shared among multiple users, represented in Braille. The added advantage of this work is , almost all the methods use cover images to hide the share while transmitting the shares, but here we are not using any cover image instead the shares will be represented in Braille. As it is difficult for an attacker to determine the language in the Braille format. Even if an attacker gets one or few shares, he can not reconstruct the image, as minimum m number of shares is required to reconstruct the image. It results in original image when the shares are used to reconstruct the image. No extra processing is required to improve the quality of the image.

References

[1]. M. Weinberg, G. Seroussi, and G. Sapiro. LOCO-I a low complexity, context-based, lossless image compression algorithm. In *Proc. DCC'96 Data compression conference*, pages 140–149, Snowbird, Utah, Mar. 1996.

[2]. M.Weinberger, J. Rissanen, and R. Arps. Applications of universal context modeling to lossless compression of gray-scale images. *IEEE Transactions on ImageProcessing*, IP-5:575–586, Apr. 1996.

[3]. R. Calderbank, I. Daubechies, W. Sweldens, and B.- L. Yeo. Lossless image compression using integer to integer wavelet transforms. In *Proc. ICIP-97, IEEEInternational Conference on Image*, volume 1, pages 596–599, Santa Barbara, California, Oct. 1997.

[4]. A. Shamir, "How to share a secret," *Communications of the ACM*, Vol. 22, 1979, pp. 612-613.

[5]. G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the AFIPS National Computer Conference*, Vol. 48, 1979, pp. 313-317.

[6]. C. C. Thien and J. C. Lin, "Secret image sharing," *Computers and Graphics*, Vol. 26, 2002, pp. 765-770.

[7]. C. P. Huang and C. C. Li, "A secret image sharing method using integer multiwavelet transform," in *Proceedings of IEEE International Conference on Image Processing*, 2006, pp. 1969-1972.

[8]. C. P. Huang and C. C. Li, "Secure and progressive image transmission through shadows generated by multiwavelet transform," *International Journal of Wavelets*, *Multiresolution and Information Processing*, Vol. 6, 2008, pp. 907-931.

[9] Young-Fu Chen , Yung-Koun Chan, Ching-Chun Huag , " A Multiple level visual secret sharing scheme without image size expansion " , Information Science 177 , 2007.

[10] Jen-Bang Feng, Hsien-Chu Wu , et al, " Visual secret sharing for multiple secrets" , Pattern Recognition 41 , 2008.

[11] Chin-Pan Huag, Chin Chang Li , " A Secret Image Sharing method using Integer Wavelet Transform" EURASIP Journal on advances in signal processing , Vol 2007.

[12] Jun Kong ,Yanfen Zang et al, " A Scalable Secret Image Sharing method based on Discret Wavelet Transformation" Springer-Verlog Berlin Heidelberg , 2007.