Embedded Visual Cryptography Schemes for Secret Images

Anandhi1[†] and S.Satthiyaraj2[††],

E.S Engineering College, Anna university, India Dr.Pauls Engineering College, Anna university, India

Summary
Visual cryptography is one of the techniques used to encrypt the images by dividing the original image into transparencies. The transparencies can be sent to the intended person, and at the other end the transparencies received person can decrypt the transparencies using the tool, thus gets the original image. The proposed Visual cryptography provides the demonstration to the users to show how encryption and decryption can be done to the images. In this technology, the end user identifies an image, which is not the correct image. That is, while transmitting the image the sender will encrypt the image using the application here sender gets the two or more transparencies of the same image. The application provides an option to the end user of encryption. The end user can divide the original image into number of different images. Using the application we can send encrypted images that are in the format of GIF and PNG. The encrypted transparencies can be saved in the machine and can be sent to the intended person by other means.

Key words:
Image processing, visual Cryptography Scheme (VCS), GIF Encoding, Decoding.

1. Introduction

Visual cryptography technology [1][4], the end user identifies an image, which is going to act as the carrier of data. The data file is also selected and then to achieve greater speed of transmission the data file and image file are compressed and sent. Prior to this the data is embedded into the image and then sent. The image if hacked or interpreted by a third party user will open up in any image previewed but not displaying the data. This protects the data from being invisible and hence is secure during transmission. The user in the receiving end uses another piece of code to retrieve the data from the image.

The scope of the System provides a friendly environment to deal with images. Generally tools supports only one kind of image formats. This application supports .gif and .png (portable network graphics) formatted images and the application has been developed using swing and applet technologies, hence provides a friendly environment to users.

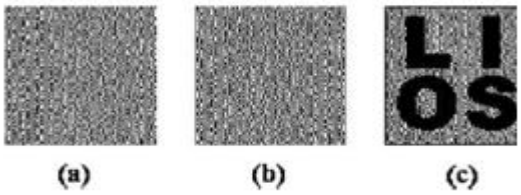

Fig.1 Example of traditional (2,2) VCS with image size 128 X 128.

The basic principle of the visual cryptography scheme (VCS) was first introduced by Naor and Shamir. VCS is a kind of secret sharing scheme that focuses on sharing secret images. The idea of the visual cryptography model proposed in is to split a secret image into two random shares (printed on transparencies) which separately reveals no information about the secret image other than the size of the secret image. The secret image [9] can be reconstructed by stacking the two shares. The underlying operation of this scheme is logical operation OR. In this paper, we call a VCS with random shares the traditional VCS or simply the VCS. In general, a traditional VCS takes a secret image as input, and outputs shares that satisfy two conditions: 1) any qualified subset of shares can recover the secret image; 2) any forbidden subset of shares cannot obtain any information of the secret image other than the size of the secret image. An example of traditional (2,2)-VCS can be found in Fig. 1, where, generally speaking, a VCS means any out of shares could recover the secret image. In the scheme of Fig. 1, shares (a) and (b) are distributed to two participants secretly, and each participant cannot get any information about the secret image, but after stacking shares (a) and (b), the secret image can be observed visually by the participants [3]. VCS has many special applications, for example, transmitting military orders to soldiers who may have no cryptographic knowledge or computation devices in the battle field. Many other applications of VCS, other than its original objective (i.e., sharing secret image), have been found, for example, authentication and identification [5], watermarking and transmitting passwords etc.

Manuscript received December 5, 2012
Manuscript revised December 20, 2012

2. Literature Survey

2.1 Visual Cryptography for General Access Structure by Multi-pixel Encoding with Variable Block Size

Multi-pixel encoding [11] is an emerging method in visual cryptography for that it can encode more than one pixel for each run. However, in fact its encoding efficiency is still low. This paper presents a novel multi-pixel encoding which can encode variable number of pixels for each run. The length of encoding at one run is equal to the number of the consecutive same pixels met during scanning the secret image. The proposed scheme can work well for general access structure and chromatic images without pixel expansion. The experimental results also show that it can achieve high efficiency for encoding and good quality for overlapped images.

2.2 Halftone Visual Cryptography

Visual cryptography [1] encodes a secret binary image (SI) into shares of random binary patterns. If the shares are xeroxed onto transparencies, the secret image can be visually decoded by superimposing a qualified subset of transparencies, but no secret information can be obtained from the superposition of a forbidden subset. The binary patterns of the shares, however, have no visual meaning and hinder the objectives of visual cryptography. Extended visual cryptography was proposed recently to construct meaningful binary images as shares using hypergraph colourings, but the visual quality [10] is poor. In this paper, a novel technique named halftone visual cryptography is proposed to achieve visual cryptography via halftoning. Based on the blue-noise dithering principles, the proposed method utilizes the void and cluster algorithm to encode a secret binary image into halftone shares (images) carrying significant visual information. The simulation shows that the visual qualities of the obtained halftone shares are observably better than that attained by any available visual cryptography method known to date.

2.3 Visual Cryptography for Print and Scan Applications

Visual cryptography is not much in use in spite of possessing several advantages. One of the reasons for this is the difficulty of use in practice [5]. The shares of visual cryptography are printed on transparencies which need to be superimposed. However, it is not very easy to do precise superposition due to the fine resolution as well as printing noise. Furthermore, many visual cryptography applications need to print shares on paper in which case scanning of the share is necessary. The print and scan process can introduce noise as well which can make the alignment difficult. In this paper, we consider the problem of precise alignment of printed and scanned visual cryptography shares. Due to the vulnerabilities in the spatial domain, we have developed a frequency domain alignment scheme. We employ the Walsh transform to embed marks in both of the shares so as to find the alignment position of these shares. Our experimental results show that the technique can be useful in print and scan applications.

2.4 Joint Visual Cryptography and Watermarking

In this paper, we discuss how to use watermarking technique for visual cryptography [5]. Both halftone watermarking and visual cryptography involve a hidden secret image. However, their concepts are different. For visual cryptography, a set of share binary images is used to protect the content of the hidden image. The hidden image can only be revealed when enough share images are obtained. For watermarking, the hidden image is usually embedded in a single halftone image while preserving the quality of the watermarked halftone image. In this paper, they proposed a joint visual-cryptography and watermarking (JVW) algorithm that has the merits of both visual cryptography and watermarking

2.5 An Improved Visual Cryptography Scheme For Secret Hiding

Visual Cryptography [7] is based on cryptography where *n* images are encoded in a way that only the human visual system can decrypt the hidden message without any cryptographic computations when all shares are stacked together. This paper presents an improved algorithm based on Chang's and Yu visual cryptography scheme for hiding a colored image into multiple colored cover images. This scheme achieves lossless recovery and reduces the noise in the cover images without adding any computational complexity.

The Visual cryptography scheme is a cryptographic technique which allows visual information (e.g. printed text, handwritten notes, and picture) to be encrypted in such a way that the decryption can be performed by the human visual system, without the aid of computers [11]. There are various measures on which performance of visual cryptography scheme depends, such as pixel expansion, contrast, security, accuracy, computational complexity, share generated is meaningful or meaningless, type of secret images (either binary or colour) and number of secret images (either single or multiple) encrypted by the scheme. The study of VCS is on the performance

analysis on the basis of pixel expansion, number of secret images, image format and type of shares generated.

3. Existing System

Visual cryptography [1] is the art and science of encrypting the image in such a way that no-one apart from the sender and intended recipient even realizes the original image, a form of security through obscurity. By contrast, cryptography obscures the original image, but it does not conceal the fact that it is not the actual image.

After generating the covering shares, the embedding process can be realized by the following algorithm.

The embedding process:

Input: The corresponding VCS (c_0, c_1) with pixel expansion and the secret image .
Output: The *n* embedded shares $e_0, e_1, ...e_{n-1}$.

Step1: Dividing the covering shares into blocks that contain ($t \geq m$) subpixels each.
Step2: Choose m embedding positions in each block in the n covering shares.
Step3: For each black (respectively, white) pixel in *I*, randomly choose a share matrix $M \in C_1$.
Step4: Embed the *m* subpixel of each row of the share matrix M into the *m* embedding positions chosen in Step2.

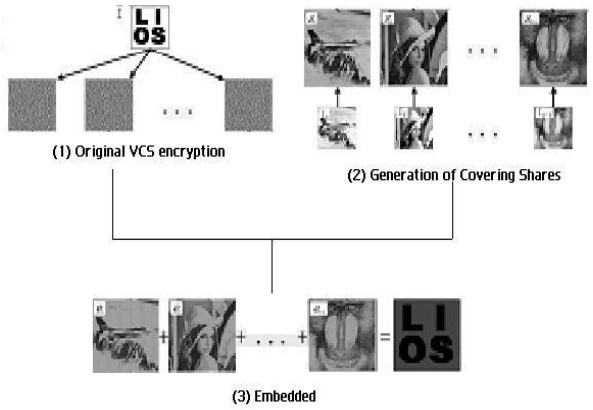

Fig.2. Embedding Process

3.1 Limitation of the Existing System

- The existing system does not provide a friendly environment to encrypt or decrypt the data (images).
- The system supports with only one type of image format only. For example, if it is .jpg, then it supports only that same kind of image format only.

4. Proposed System

Proposed system Visual cryptography provides a friendly environment to deal with images. Generally cryptography tools supports only one kind of image formats. The application supports .gif and .png (portable network graphics) formatted images and the application has been developed using swing and applet technologies, hence provides a friendly environment to users.

4.1 LZW Data Compression Algorithm

Lempel–Ziv–Welch (LZW) is a universal lossless data compression algorithm created by Abraham Lempel, Jacob Ziv, and Terry Welch. It was published by Welch in 1984 as an improved implementation of the LZW algorithm published by Lempel and Ziv[13] in 1978. The algorithm is simple to implement, and has the potential for very high throughput in hardware implementations.

The scenario described in Welch's 1984 paper [1] encodes sequences of 8-bit data as fixed-length 12-bit codes. The codes from 0 to 255 represent 1-character sequences consisting of the corresponding 8-bit character, and the codes 256 through 4095 are created in a dictionary for sequences encountered in the data as it is encoded. At each stage in compression, input bytes are gathered into a sequence until the next character would make a sequence for which there is no code yet in the dictionary. The code for the sequence (without that character) is emitted, and a new code (for the sequence with that character) is added to the dictionary.

The idea was quickly adapted to other situations. In an image based on a color table, for example, the natural character alphabet is the set of color table indexes, and in the 1980s, many images had small color tables (on the order of 16 colors). For such a reduced alphabet, the full 12-bit codes yielded poor compression unless the image was large, so the idea of a variable-width code was introduced: codes typically start one bit wider than the symbols being encoded, and as each code size is used up, the code width increases by 1 bit, up to some prescribed maximum (typically 12 bits).

Further refinements include reserving a code to indicate that the code table should be cleared (a "clear code", typically the first value immediately after the values for the individual alphabet characters), and a code to indicate the end of data (a "stop code", typically one greater than the clear code). The clear code allows the table to be reinitialized after it fills up, which lets the encoding adapt to changing patterns in the input data. Smart encoders can monitor the compression efficiency and clear the table whenever the existing table no longer matches the input well.

Since the codes are added in a specific manner which determined by the data, the decoder mimics the building table as it sees the resulting codes. It is critical that the encoder and decoder agree on which variety of LZW is being used: the size of the alphabet, the maximum code width, whether variable-width encoding is being used, the initial code size, whether to use the clear and stop codes (and what values they have). Most formats that employ LZW build this information into the format specification or provide explicit fields for them in a compression header for the data.

4.2 Process of LZW Algorithm

The proposed systems use the LZW (Lempel-Ziv-Welch) Algorithm. The method used to implement in the following process.

1. Select the gray scale image.
2. Apply the LZW compression technique for the gray scale image.
3. Preparing the dictionary for the gray scale images.
4. In dictionary replaces strings of characters with Single codes.
5. Calculations are done by dynamic Huffman coding.
6. In compression of greyscale image select the secret Information pixels.
7. Then generation halftone shares using error diffusion Method.
8. Filter process is applied for the output gray scale images.

Filters are used to improve the quality of reconstructed image to minimize the noises for sharpening the input secret image.

4.3 Uses

LZW [13] compression became the first widely used universal data compression method on computers. A large English text file can typically be compressed via LZW to about half its original size.

LZW was used in the public-domain program compress, which became a more or less standard utility in Unix systems circa 1986. It has since disappeared from many distributions, both because it infringed the LZW patent and because gzip produced better compression ratios using the LZ77-based DEFLATE algorithm, but as of 2008 at least FreeBSD includes both compress and uncompress as a part of the distribution. Several other popular compression utilities also used LZW, or closely related methods.

LZW became very widely used when it became part of the GIF image format in 1987. It may also (optionally) be used in TIFF and PDF files. (Although LZW is available in Adobe Acrobat software, Acrobat by default uses DEFLATE for most text and color-table-based image data in PDF files.)

4.4 Advantages of Proposed System

- The Embedded Visual Cryptography Schemes for Secret images tool is easy to use.
- The image are compressed and send the receiver in order to decrease the size and for fast transmitting the data (image)
- It support .gif and .png formats only.
-

5. Module Description

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

5.1. Interface Design using Applet frame work

In this module, we design user interface design using applet frame work. The user interface should be very easy and understandable to every user. So that anyone can access using the system. It must be supportable using various GUIs. The user interface also consists of help file. The help file assists on every concepts of the embedded visual cryptography. Help file should clearly depict the details of the project developed in simple language using various screen shoots.

5.2 Visual Cryptography

This module is the core for the project, where we implement the Visual Cryptography. We used LZW Data Compression algorithm. The LZW data compression algorithm is applied for the gray scale image here. As a pre-processing step, a dictionary is prepared for the gray scale image. In this dictionary, the string replaces characters with single quotes. Calculations are done using dynamic Huffman coding. In compression of greyscale image select the information pixels. Then generate halftone shares using error diffusion method. At last filter process is applied for the output gray scale images. Filters are used to improve the quality of reconstructed image to minimize the noises for sharpening the input secret image.

5.3 Encoding

A high level view of the encoding algorithm is shown here:

1. Initialize the dictionary to contain all strings of length one.

2. Find the longest string W in the dictionary that matches the current input.
3. Emit the dictionary index for W to output and remove W from the input.
4. Add W followed by the next symbol in the input to the dictionary.
5. Go to Step 2.

A dictionary is initialized to contain the single-character strings corresponding to all the possible input characters (and nothing else except the clear and stop codes if they're being used). The algorithm works by scanning through the input string for successively longer substrings until it finds one that is not in the dictionary. When such a string is found, the index for the string less the last character (i.e., the longest substring that is in the dictionary) is retrieved from the dictionary and sent to output, and the new string (including the last character) is added to the dictionary with the next available code. The last input character is then used as the next starting point to scan for substrings.

In this way, successively longer strings are registered in the dictionary and made available for subsequent encoding as single output values. The algorithm works best on data with repeated patterns, so the initial parts of a message will see little compression. As the message grows, however, the compression ratio tends asymptotically to the maximum.

5.4 Decoding

The decoding algorithm works by reading a value from the encoded input and outputting the corresponding string from the initialized dictionary. At the same time it obtains the next value from the input, and adds to the dictionary the concatenation of the string just output and the first character of the string obtained by decoding the next input value. The decoder then proceeds to the next input value (which was already read in as the "next value" in the previous pass) and repeats the process until there is no more input, at which point the final input value is decoded without any more additions to the dictionary.

In this way the decoder builds up a dictionary which is identical to that used by the encoder, and uses it to decode subsequent input values. Thus the full dictionary does not need be sent with the encoded data; just the initial dictionary containing the single-character strings is sufficient (and is typically defined beforehand within the encoder and decoder rather than being explicitly sent with the encoded data.)

5.5 Creating Transparencies

This scheme provides theoretically perfect secrecy. An attacker who obtains either the transparency image or the screen image obtains no information at all about the encoded image since a black-white square on either image is equally likely to encode a clear or dark square in the original image. Another valuable property of visual cryptography is that we can create the second layer after distributing the first layer to produce any image we want. Given a known transparency image, we can select a screen image by choosing the appropriate squares to produce the desired image. One of the most obvious limitations of using visual cryptography in the past was the problem of the decoded image containing an overall gray effect due to the leftover black sub pixel from encoding. This occurred because the decoded image is not an exact preproduction, but an expansion of the original, with extra black pixel. Black pixel in the original document remains black pixel in the decoded version, but White pixel becomes gray. This resulted in a lot of contrast to the entire image. The extra black sub pixel in the image causes the image to become distorted.

D - Secret information. K - Number of shares generated from D. share - piece of information.

Divide data D into n pieces in such a way that D is easily reconstruct able from any k pieces, but even complete knowledge of any k-1 pieces reveals no information about D. Stacking two pixels (each consists of four sub-pixels) can occur for example the following two cases: Secret sharing scheme is a method of sharing secret information among a group of participants. In a secret sharing scheme, each participant gets a piece of secret information, called a share. When the allowed coalitions of the participants pool their shares, they can recover the shared secret; on the other hand, any other subsets, namely non-allowed coalitions, cannot recover the secret image by pooling their shares. In the last decade, various secret sharing schemes were proposed, but most of them need a lot of computations to decode the shared secret information.

The basic 2 out of 2 visual cryptography model consist of secret message encoded into two transparencies, one transparency representing the cipher text and the other acting as a secret key. Both transparencies appear to be random dot when inspected individually and provide no information about the original clear text. However, by carefully aligning the transparencies, the original secret message is reproduced. The actual decoding is accomplished by the human visual system. The original is encrypted into 2 transparencies you need both transparencies to decode the message.

5.6 Un-hiding Image from Transparency

The simplest form of visual cryptography separates an image into two layers so that either layer by itself conveys no information, but when the layers are combined the image is revealed. One layer can be printed on a

transparency, and the other layer displayed on a monitor. When the transparency is placed on top of the monitor and aligned correctly, the image is revealed. For each image pixel, one of the two encoding options is randomly selected with equal probability. Then, the appropriate colorings of the transparency and screen squares are determined based on the color of the pixel in the image.

6. Conclusions

The Embedded visual cryptography scheme tool is simple and easy to use. Various visual cryptography schemes are studied and their performance is evaluated on four criteria: number of secret images, pixel expansion, image format and type of share generated. Security is the primary concern of today's communication world, is successfully implemented using the IDEA algorithm. It provides a safe and secure transmission as it involves multiple manipulations for encryption and so is it with decryption. This System provides a friendly environment to deal with images. Generally tools supports only one kind of image formats.

This application supports .gif and .png (portable network graphics) formatted images and the application has been developed using swing and applet technologies, hence provides a friendly environment to users.

References

[1] Feng Liu and chuankun Wu.(2011), 'Embedded Extended Visual Cryptography Schemes', IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 6, NO. 2, pp. 307-322
[2] Shamir A. (1979), 'How to share a secret' Commun. ACM, vol. 22, no. 11, pp. 612–613.
[3] Blakley G. R. (1979), 'Safeguarding cryptographic keys' in Proc. National Computer Conf., vol. 48, pp. 313–317.
[4] Naor M. and Shamir A. (1995), 'Visual cryptography' in Proc. EUROCRYPT 94, Berlin, Germany, vol. 950, pp. 1–12, Springer-Verlag, LNCS.
[5] Naor M. and Pinkas B. (1997), 'Visual authentication and identification' in Proc. CRYPTO'97, vol. 1294, pp. 322–336, Springer-Verlag
[6] Chen T. H. and Tsai D. S. (2006), 'Owner-customer right protection mechanism using a watermarking scheme and a watermarking protocol' Pattern Recognit., vol. 39, pp. 1530–1541.
[7] Tuyls P., Kevenaar T., Schrijen G. J., Staring T., and Van Dijk M. (2004), 'Security displays enabling secure communications' in Proc. First Int. Conf. Pervasive Computing, Boppard Germany, Springer-Verlag Berlin LNCS, vol. 2802, pp. 271–284.
[8] Blundo C., De Bonis A., and De Santis A. (2001), 'Improved schemes for visual cryptography' Designs, Codes and Cryptography, vol. 24, pp. 255–278.
[9] Ateniese G., Blundo C., De Santis A., and Stinson D. R. (1996), 'Visual cryptography for general access structures' Inf. Computat., vol. 129, pp. 86–106.
[10] Prakash N. K. and Govindaraju S. (2007), 'Visual secret sharing schemes for color images using halftoning' in Proc. Int. Conf. Computational Intelligence and Multimedia Applications (ICCIMA 2007), vol. 3, pp. 174–178.
[11] Luo H., Yu F.X., Pan J. S., and Lu Z. M. (2008), 'Robust and progressive color image visual secret sharing cooperated with data hiding' in Proc. 2008 Eighth Int. Conf. Intelligent Systems Design and Applications, vol. 3, pp. 431–436.
[12] Hou Y. C. (2003), 'Visual cryptography for color images' Pattern Recognit., vol. 1773, pp. 1–11.
[13] http://en.wikipedia.org/wiki/Lempl-Ziv-Welch
[14] http://www.sourcefordgde.com.
[15] http://www.ieice.org/eng/shiori/mokuji.html

Anandhi received the degree in Computer Science in the year 2006 and did her post graduation in Computer Applications in 2009 from Pondicherry University .And working as a senior lecturer in the department of computer applications in E S college of engineering, India.

S.Satthiyaraj received his diploma in EEE in the year 2003.And did his B.E and M.E in EEE from anna university in 2006 & 2011.Currently working as a assistant Professor in Dr. Pauls Engineering College, India