Prashant Gandhi *et al*, International Journal of Computer Science and Mobile Applications,
Vol.2 Issue. 12, December- 2014, pg. 30-37 **ISSN: 2321-8363**

Embedded Extended Visual Cryptography Schemes

Prashant Gandhi, Pushpendra Gautam, Rohit Bisht
Scholar, Department of Computer Science and Engineering, AMITY University, Greater Noida, U.P., INDIA
Shubhi Gupta
Assistant Professor, Department of Computer Science and Engineering, AMITY University, Greater Noida, U.P., INDIA

ABSTRACT

A visual cryptography scheme (VCS) is a kind of secret image sharing scheme which allows the encoding of a secret image into shares. The drawback of visual cryptography scheme (VCS) is that even a layman to cryptography is able to decode the secret image without having any cryptographic knowledge and computational tools/devices. An extended visual cryptography scheme (EVCS) is a type of visual cryptography scheme (VCS) which consists of meaningful shares (not like in the traditional VCS which consists of meaningful as well as meaningless shares). In this paper, we propose the construction of EVCS which is recognized by embedding random shares into meaningful covering shares, and hence we call it the Embedded Extended Visual Cryptography Scheme (EEVCS). Experimental results in recent years show that the proposed Embedded EVCS has reasonably good visual quality compared to many of the well-known EVCSs.

INDEX TERMS:

Image processing, Visual Cryptography Scheme (VCS), GIF Encoding, Decoding.

1. INTRODUCTION

A visual cryptography scheme (VCS) is a kind of secret image sharing scheme which allows the encoding of a secret image into shares. The drawback of visual cryptography scheme (VCS) is that even a layman to cryptography is able to decode the secret image without having any cryptographic knowledge and computational tools/devices. An extended visual cryptography scheme (EVCS) is a type of visual cryptography scheme (VCS) which consists of meaningful shares (not like in the traditional VCS which consists of meaningful as well as meaningless shares). In this paper, we propose the construction of EVCS which is recognized by embedding random shares into meaningful covering shares, and hence we call it the Embedded Extended Visual Cryptography Scheme (EEVCS). Experimental results in recent years show that the proposed Embedded EVCS has reasonably good visual quality compared to many of the well-known EVCSs.
Systems provide approachable environment to deal with images. Mostly tools support only few types of image formats. Our application supports .gif and .png (portable network graphics) formatted images and the application has been developed using swings and applets.

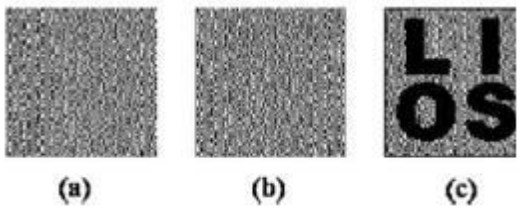

Fig.1 Example of traditional VCS

The basic norm of the visual cryptography scheme (VCS) was first introduced by Naor and Shamir. VCS is a type of secret sharing scheme that emphases on sharing secret images. The general idea of the visual cryptography model proposed is to split a secret image into two or more random shares which do not reveal any information about the secret image except the size. The secret image is reconstructed by heaping up the shares. The core operation of this scheme is to perform logical OR operation.
VCS has many different applications, for example, transmitting military orders to soldiers who may have no cryptographic knowledge or computation devices in the battle field, authentication and identification, watermarking and transmitting passwords etc.

2. LITERATURE SURVEY

2.1 Visual cryptography for general access structure by multi-pixel encoding with variable block size.

Multi-pixel coding is an evolving technique of visual cryptography which encodes more than one pixel every time it is executed. However, its encoding efficiency is still low. The size of encoding in single run is equal to the number of similar successive pixels met during the scanning of secret image. The proposed system works well for general access to structures and chromatic images without the pixel being amplified. The experimental results show that high efficiency for encoding and good quality for overlapped images can be achieved.

2.2 Halftone visual cryptography

The visual cryptography failed to work with the gray scale images, so to overcome this drawback Halftone technique was introduced. The halftone technique is also known as dithering technique. The biggest advantage of halftone technique is that it coverts a gray scale image into a binary image. This technique uses certain amount of black and white pixels in the form of patterns in order to achieve the gray scale. The percentages of black and white pixels represent different grayness.

The basic principle of halftoning is to map the pixels of gray scale from the original image into the black pixels with patterns. However, this process requires a lot of memory space. So, to overcome this problem, we use dithering matrix which is a kind of integer matrix.
The halftoning process is described below:

Algorithm : The halftoned process :

Input: The dithering matrix and a pixel with gray scale in input image
Output: The halftoned pattern at the position of the pixel
For i = 0 to a -1 do
For j = 0 to b - 1 do

If f≤ Dij then print a black pixel at position (i ,j);
else print a white pixel at position (i ,j);

2.3 Visual cryptography for scan & print applications

Visual cryptography has several advantages but is not used in a manner it should be because of the reason that it is the difficult to use. The shares of visual cryptography are copied on transparencies which is to be superimposed but it is not at all easy to precisely superposition the transparencies due to the fine resolution and the printing noise. To print shares on paper through which case scanning of the share can be done requires many visual cryptography applications. The disadvantage of print and scan process is that it can introduce noise which can affect the alignment. In this paper, we contemplate the problem of precise alignment of printed and scanned visual cryptography shares. Because of the vulnerabilities in the spatial domain, we have developed a frequency domain alignment scheme in which we employ the Walsh transform to embed marks in both of the shares so as to find the alignment position of these shares. The 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 confer that how watermarking technique be used by us for visual cryptography. Both the previously discussed techniques, they are, halftone watermarking and visual cryptography involve a hiding secret image but their concepts are different. In visual cryptography, a bunch of share binary images is used to protect the content of the secret hidden image. The hidden image can only be revealed when enough share images are obtained. For watermarking, thehidden image is usually entrenched in a single halftone image while preserving the quality of the watermarked halftone image. In this paper, we proposed a joint visual-cryptography and watermarking (JVW) algorithm which has the qualities of both visual cryptography and watermarking.

2.5 An improved visual cryptography scheme for secret hiding

Visual Cryptography is based on cryptography where ‘n’ images are encoded in such a way that only the human visual system can decrypt the hidden message without the use of any cryptographic tools when all shares are arranged 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 attains 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 cryptographic computations/tools. There are different measures on which the performance of visual cryptography scheme depends, like pixel expansion, contrast, security, accuracy, computational complexity, share generated is meaningful or meaningless, type of secret images (either binary or color) and number of secret images (either single or multiple) encrypted by the scheme.

3. PREVAILING SYSYTEM

Visual cryptography is the technique for encrypting an image in such a way that no-one apart from the sender and the intended receiver can even realize the original image, i.e., a form of security through anonymity. Cryptography disguises the original image, but it does not hide the fact that the image is not the actual image. After the generation of covering shares, the embedding procedure can be understood by the following algorithm:

The embedding procedure:

Input: 'n' covering shares, the corresponding VCS (C0,C1) with pixel expansion 'm' and secret image 'i'.

Output: The 'n'embedded shares a0,a1, ...an-1.

Step1: Divide the covering shares into separate blocks thatcontain t (≥ m) sub pixels;

Step2: For all odd p ∈ {0,.....,k}, call $\mathrm{ADD}(\mathrm{p}, \mathrm{M}')$;

Step3: Choose m embedding positions at each block in the n covering shares;

Step4: For each black (respectively, white) pixel in i , randomly choose a share matrix M ∈ $\mathrm{C1}$ $(\mathrm{respectively}\ , \mathrm{M} \in \mathrm{C0})$;

Step5: Embed the 'm' sub pixel of each row of the share matrix 'M' into the 'm' embedding positions chosen in Step2.

3.1 Limitations of the Prevailing System

- The prevailing system does not provide friendly environment to encrypt or decrypt the images.
- Generally the systems support only one type of image format. For example, if it is .jpg, then the system will support only that similar kind of image format.

4. PROJECTED SYSTEM

Projected system of visual cryptography provides a friendly environment to pact with the images. Usually cryptographic tools support only one kind of image formats but this application supports different formats like .gif and .png (portable network graphics).

4.1 LZW Data Compression Algorithm

Lempel–Ziv–Welch (LZW) is a common lossless data compression algorithm designed by Abraham Lempel, Jacob Ziv and Terry Welch. This algorithm was published by Welch in the year 1984 as an enhanced implementation of the previously given LZW algorithm which was published by Lempel and Ziv

in 1978. The algorithm is easy to implement, and has potential for very high throughput in hardware implementations.
The scenario described in Welch's 1984 paper 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 the sequences encountered in the data as it is encoded. At each stage during the compression, input bytes are gathered into a sequence until the next character makes a sequence for which there is no code yet. The code for the sequence is emitted, and a new code (for the sequence with that character) is added to the dictionary. In an image based on color table, for example, the natural character alphabet is a set of color table indexes, and during the 1980s, many images had small color tables (in the order of 16 colors). For such a reduced alphabet, the full 12-bit codes produced poor compression until and unless the image was large, so the idea of variable-width code was presented which stated: 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. Further modifications 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 re-modified after it seals up, which lets the encoding adapt to the 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 is determined by the data, the decoder copies 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 should 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 is implemented in the following process:

1. Select gray scale image.
2. Apply LZW compression technique for the gray scale image.
3. Prepare a dictionary for the gray scale images.
4. In dictionary, substitute the string of characters with single codes.
5. Calculations are done by dynamic Huffman coding.
6. During compression of gray scale image select the secret information pixels.
7. Then generate halftone shares using error diffusion method.
8. Filter process is applied for the output of the gray scale images.

Filters are used to improve the quality of recreated images to diminish the noise for sharpening the input secret image.

4.3 Uses

LZW compression technique became the very first commonly used data compression technique on computers. LZW was used in the public-domain program compress, which became less or more standard utility in Unix gzip DEFLATE compress uncompressed systems circa 1986. Since then it has disappeared from many distributions, both because it infringed the LZW patent and because produced better compression ratios using the LZ77-based algorithm, but as of 2008 at least FreeBSD includes both. Several other popular compression services also used LZW, or narrowly related methods. LZW became

very widely used when it became part of the GIF TIFF PDF Adobe Acrobat DEFLATE image format in 1987. It can also be used in files.

4.4 Advantages of proposed system

• The Embedded Visual Cryptography Schemes (EVCS) for Secret images tool is easy to use.
• The images are compressed and sent to the receiver in order to cut the size and for fast transmission of the data (image).
• It ropes .gif and .png formats only.

5. MODULE DESCRIPTION

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

5.1. Interface design using applet frame work

In this module, we design user interface (UI) design using the applets. The user interface (UI) should be easy and clear to all the users so that everyone is able to gain access using the system. It must be supportable by various GUIs. The UI consists of help files. The help files assists on every concept of the embedded visual cryptography. Help files clearly show the particulars of the project developed in simple language using various screen shoots.

5.2 Visual cryptography

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

5.3 Encoding

A great 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. Release 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. Then go to Step 2

A dictionary is initialized to contain single-character strings corresponding to all the possible input characters (except the clear and stop codes if they're being used). The algorithm works by scanning the input string for successively longer substrings until and unless it finds one that is not in the dictionary. When such a string is found, the index of the string for the last character (i.e., the longest substring that is

in the dictionary) is retrieved from the dictionary and sent to output, and a 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 that the initial parts of the message see little compression. As the message gets bigger, however, the compression ratio inclines asymptotically to the maximum.

5.4 Decoding

The decoding algorithm works by reading a value from the encoded input and accordingly outputs the corresponding string from the modified dictionary. At the same time it obtains the next value from the input, and adds to the dictionary of the concatenation of the string the 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 are no more inputs, at which the final input value is decoded without any more additions to the dictionary.
In this way the decoder builds up a dictionary which is indistinguishable to that used by the encoder, and uses it to decode the subsequent input values. Thus the full dictionary does not need to 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 theoretically provides faultless secrecy. An unauthorized user who obtains either of the transparency images or the screen image, acquires no information at all about the encoded image. Another cherished 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 selecting appropriate squares to produce the desired images. One of the perceptible limitations of using visual cryptography was the problem of the decoded image containing an overall gray effect due to the leftover black sub pixel from encoding. This followed because the decoded image is not an exact preproduction, however only an expansion of the original, with extra black pixel. Black pixel in the original document remains the same (i.e., black pixel) in the decoded version too, but the white pixel converts gray. This results 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 the 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 data D. Stacking two pixels (each consisting 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 claimants. In a secret sharing scheme, every applicant gets a piece of secret information, called a share. When the allowed coalitions of the participants combine their shares, they can recover the shared secret; and on the other hand, any other subset, namely non-allowed coalitions, cannot recover the secret image by combining their shares.

In the last 10 years, 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 consists of secret message encoded in two transparencies, one transparency represents the cipher text and the other represents a secret key. Both transparencies appear to be random dot when reviewed individually and gives no information about the original clear text. However, after carefully aligning the transparencies, the original secret message is generated. The actual decoding is accomplished by the human optical system. The original message is encrypted into 2 transparencies for which you need both the transparencies to decode the message.

5.6 Un-hiding image from transparency

The simplest form of visual cryptography splits an image into two layers so that either of the layers conveys no information about themselves, but when the layers are combined only then the image is revealed. One of the layers can be printed on a transparency and the other layer is displayed on a monitor. When the transparency is placed on top of the monitor screen and associated correctly, then only the image is revealed. For every 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 (EVCS) tool is simple and is easy to practice. Different visual cryptography schemes are studied and their performances are evaluated on the basis of four basic criteria's: 1) Number of secret images, 2) Pixel expansion, 3)Format of the image and 4)Type of transparency generated. Security is the foremost concern of present communication world and is successfully executed using the IDEA algorithm. It provides a safe and secure transmission as it involves number of manipulations for encryption and decryption. Usually tools support only one kind of image format but this application supports .gif and .png (portable network graphics) formatted images and the application has been developed using swing and applet which provides a friendly environment to the users of the application.

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} http://www.sciencedirect.com/science/article/pii/S0890540196900760

{3} http://www.slideshare.net/nanonsrc/embedded-extended-visual-cryptography-schemes

