Abstract
A visual cryptography scheme (VCS) is a kind of secret sharing scheme which allows the encoding of a secret image into shares distributed to participants. The beauty of such a scheme is that a set of qualified participants is able to recover the secret image without any cryptographic knowledge and computation devices. An extended visual cryptography scheme (EVCS) is a kind of VCS which consists of meaningful shares (compared to the random shares of traditional VCS). In this paper, we propose a construction of EVCS which is realized by embedding random shares into meaningful covering shares, and we call it the embedded EVCS. Experimental results compare some of the well-known EVCSs proposed in recent years systematically, and show that the proposed embedded EVCS has competitive visual quality compared with many of the well-known EVCSs in the literature. In addition, it has many specific advantages against these well-known EVCSs, respectively.

OUR PROPOSED ABSTRACT:
Visual cryptography is one of the technique 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 our tool, thus gets the original image. Our 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 our application here sender gets the two or more transparencies of the same image. Our application provides an option to the end user of encryption. The end user can divide the original image into number of different images. Using our 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.

**Scope:**

System provides a friendly environment to deal with images. Generally tools support only some kinds of image formats. Our application supports .gif and .png (portable network graphics) formatted images and our application has been developed using swing and applet technologies, hence provides a friendly environment to users.

**Existing System:**

**Visual cryptography** 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.

**Limitation:** - The existing system does not provide a friendly environment to encrypt or decrypt the data (images).

**Proposed System:**

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

**Problem Definition:** When ever we transmit the data(image) in the network, any unauthenticated person can read our data(image). In order to provide security to data(image) generally sender will encrypt the data(image) and send it the intended person and the receiver will decrypt the encrypted data(image) and uses it.

**Hardware Requirements:**
- Processor: Intel or AMD processor computer
- RAM: 256 MB or more
- Hard Disk Space: 8 GB or more

**Software Requirements:**
- Operating System: Windows XP
- Technologies: JAVA 6.0, Swing
- Tools: NetBeans 6.0

**MODULES :-**

- Interface design using Applet frame work
- visual cryptography implementation
- Testing and integration

**Base paper Algorithm:**

**Input:** The c x d dithering matrix D and a pixel with gray-level g in input image I. 

**Output:** The halftoned pattern at the position of the pixel 

For i=0 to c-1 do
For j=0 to d-1 do
If g<=Dij then print a black pixel at position (i,j);
Else print a white pixel at position (i,j);

For embedding

**Input:** The \( n \) covering shares constructed in Section IV, the corresponding VCS \((C_0, C_1)\) with pixel expansion \( m \) and the secret image \( I \).

**Output:** The \( n \) embedded shares \( e_0, e_1, \ldots, e_{n-1} \).

Step 1: Dividing the covering shares into blocks that contain \( t(\geq m) \) subpixels each.
Step 2: Choose \( m \) embedding positions in each block in the \( n \) covering shares.
Step 3: For each black (respectively, white) pixel in \( I \), randomly choose a share matrix \( M \in C_1 \) (respectively, \( M \in C_0 \)).
Step 4: Embed the \( m \) subpixels of each row of the share matrix \( M \) into the \( m \) embedding positions chosen in Step 2.

**Our Proposed Algorithm:**

**LZW Data Compression 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 a Single codes.
5. Calculations is done by dynamic Huffman coding.
6. In compression of grayscale image select the secret Information pixels.
7. Then generation haftone shares using error diffusion Method.
8. Filter process is apply 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.

**MODULES :-**

- Interface design using Applet framework
- Visual cryptography implementation
- Testing and integration

**MODULES DESCRIPTION:**

**Interface design using Applet framework**
In this module, we design user interface design using applet framework. The user interface should be very easy and understandable to every user. So that any one can access using our system. It must be supportable using various GUIs. The user interface also consist 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.

**Visual cryptography Implementation**
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 preprocessing step, a dictionary is prepared for the gray scale image. In this dictionary, the strings 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. Atlast 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.

**Testing and integration**
This is the final module, which consists of integration of Visual cryptography implementation module into interface design using applet viewer. Then we need to test with various images and formation of transparencies. The transparencies should be able to save and load into the user interface.

Class Diagram:--

Reference: