Information and Web Security Project Methodology

Data Hiding using Image Encryption in MATLAB

GROUP MEMBERS:

First Name	Last Name	Registration Number	Roll Number
Suvimal	Yashraj	140911394	51
Ashray	Dimri	140911038	06
Riju	Khatri	140911088	14
Rishabh	Drolia	140911316	39
Aman	Chopra	140911358	44

Synopsis:

In our project we have used methods to hide text and images in an image which include, encryption of the image, embedding of data and extraction of data. This consists of two phases, the encryption phase and the decryption phase. Different techniques have been used for encrypting text and image respectively.

The objective of this project is to provide an efficient data hiding technique with minimal distortion of the encrypting image using Image Encryption in which data and image can be retrieved independently.

Information and Web Security Project Methodology

Methodology:

The program can Encode Text into Image, Decode Text from Image, Encrypt image into another image and decrypt image from the encrypted image.

Algorithm:

- Hiding an image inside the other:
- Only two bit in pixel of the image 1 is affected
- Image2 is split & store in two diagonally opposite quadrants
- The first image is resize to double its original size
- Logically the image is divided to four partitions
- The bits of the image to be hid is stored in the first image as shown below:

```
|-------------|
|d7,d6 |d3,d2 |
|-------------|
|d1,d0 |d5,d4 |
|_____________ |
```

Encrypting Text

The first part of the project is hiding text using Image Encryption. Here, the text to be hidden is written in a text file. The maximum number of characters that can be hidden in the image is equal to the product of width and height of the image, in pixels. The image in first converted into binary values and stored into a matrix. Then the text in the file is also converted to binary values and stored in a matrix. After that the corresponding binary values of the two matrices are added, and this is how data encryption has been performed.

Decrypting Text

The second part of the project is decrypting the text hidden in an image. For decryption, the encrypted image and the original image are compared, and the corresponding binary values are stored in a matrix, and then converted to text and written in another file.

Encrypting Image

The third part is encrypting an image into another image. This can be done by increasing the size of image 1(image used to hide image 2) then encoding pixel details of image 2 into image 1. By doing this the quality of image 2 will not be affected. Also only 2 bits in every pixel of image 1 is used for encoding this is because the encoded image does not show any patches of image 2. The bits in every pixel of the image 2 is split into four and placed in four different locations in image 1. So it provides some encryption. So it will be hard for others to decode the hidden image.

Below are the detailed methods used for the same:

Inverse Data Hiding: The inverse data hiding scheme is used for encrypting the images. It is mainly used to embed additional message into some distortion, with an inverse manner so that the original content can be perfectly restored after extraction of the hidden messages. The proposed scheme is the content owner encrypts the original uncompressed image using an encryption key to produce an encrypted

image. The data embedding algorithm hides data into the encrypted image using (least significant bit) LSB method with hidden key. According to the data-hiding key, he can further extract the embedded data and recover the original image from the decrypted version. Within an encrypted image containing additional data, a receiver must first decrypt it using the encryption key and the decrypted version is similar to the original image.

The detailed instructions are as follows:

1. Determine an image, which is determined as a cover image.
2. Convert the cover image to binary.
3. Random bits are generated which act as the encryption key.
4. XOR operation is performed between the random bits generated and cover image.
5. Convert the above binary image to greyscale image. Thus, encrypted image is obtained.

LSB method: This method is used to hide the data in encrypted images. In 8-bit gray scale images are selected as the cover media. Cover-images with the hidden messages embedded in them are called stego-images. For data hiding methods, the image quality refers to the quality of the stego-images. One of the common techniques is based on manipulating the least-significant-bit (LSB) planes by directly replacing the LSBs of the cover image with the message bits. LSB methods typically achieve high capacity. This allows a person to hide data in the cover image and make sure that no human could detect the change in the cover image. The LSB method usually does not increase the file size, but depending on the size of the data that is to be hidden inside the file, the file can become noticeably distorted.

The detailed procedures as follows:

1. Determine an encrypted image (cover image).
2. Determine the message or data that should be hidden into the cover image.
3. Convert all the pixel values of the encrypted image from grayscale to 8-bit binary values.
4 Embed the message or data into the cover image by hiding the data into the LSB bit of cover image.
5. The original image along with the data is recovered with less distortion and the PSNR value is calculated.

Decrypting the image

For decrypting the image, the encrypted image's resolution is decreased and made the same as the original image's resolution. Then the matrix that had been formed while encrypting is used as a key to decrypt the image.

Figure 1 below shows the system architecture followed in this project.

Information and Web Security Project Methodology

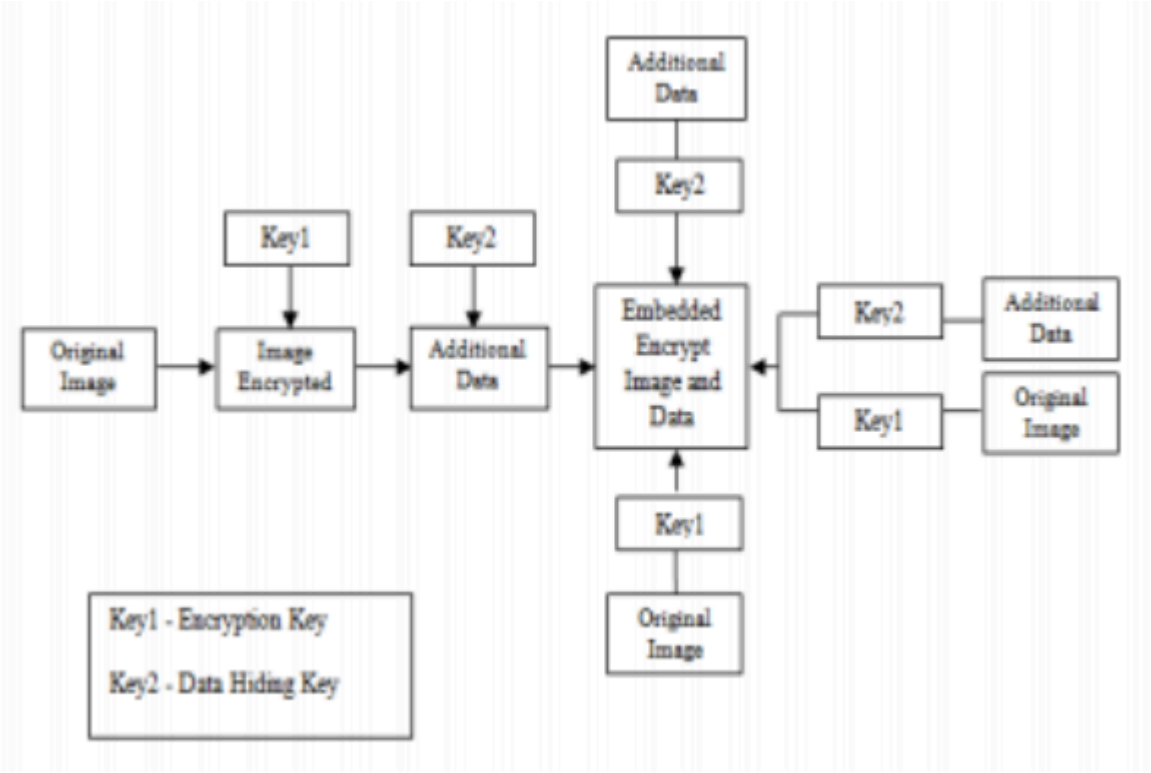

Fig.1. System Architecture

References:

http://www.arpnjournals.com/jeas/research_papers/rp_2014/jeas_0514_1083.pdf
http://www.ijircce.com/upload/2014/march/32_Inverse.pdf