Separable Reversible Data Hiding in Encrypted Image

Vanita D.Jadhav

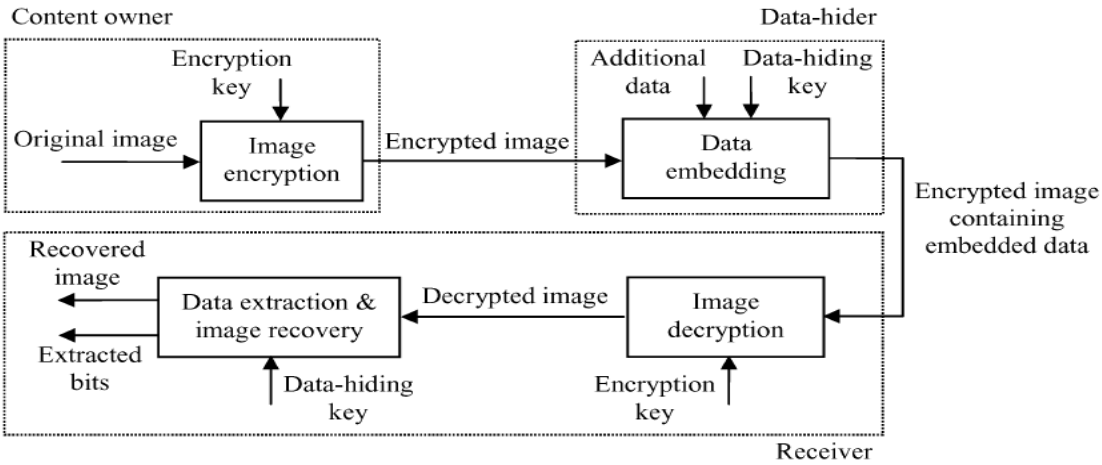
Department of Computer Science and Engineering, SVERI's College of Engineering, Pandharpur

Faculty Member

ABSTRACT:

This work proposes a novel scheme for separable reversible data hiding in encrypted images. In the first phase, a content owner encrypts the original uncompressed image using an encryption key. Then, a data-hider may compress the least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodate some additional data. With an encrypted image containing additional data, if a receiver has the data-hiding key, he can extract the additional data though he does not know the image content. If the receiver has the encryption key, he can decrypt the received data to obtain an image similar to the original one, but cannot extract the additional data. If the receiver has both the data-hiding key and the encryption key, he can extract the additional data and recover the original content without any error by exploiting the spatial correlation in natural image when the amount of additional data is not too large.

ARCHITECTURE:



EXISTING SYSTEM:

In some applications, an inferior assistant or a channel administrator hopes to append some additional message, such as the origin information, image notation or authentication data, within the encrypted image though he does not know the original image content. some parameters are embedded into a small number of encrypted pixels, and the of the other encrypted pixels are compressed to create a space for accommodating the additional data and the original data at the positions occupied by the parameters.

PROPOSED SYSTEM:

The proposed scheme is made up of image encryption, data embedding and data-extraction/image-recovery phases. The content owner encrypts the original uncompressed image using an encryption key to produce an encrypted image. Then, the data-hider compresses the least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodate the additional data. At the receiver side, the data embedded in the created space can be easily retrieved from the encrypted image containing additional data according to the data-hiding key. Since the data embedding only affects the LSB, a decryption with the encryption key can result in an image similar to the original version. When using both of the encryption and data-hiding keys, the embedded additional data can be successfully extracted and the original image can be perfectly recovered by exploiting the spatial correlation in natural image.

MODULES:

1. **Image Encryption:**

The reversible data hiding in encrypted image is investigated in. Most of the work on reversible data hiding focuses on the data embedding/extracting on the plain spatial domain. But, in some applications, an inferior assistant or a channel administrator hopes to append some additional message, such as the origin information, image notation or authentication data, within the encrypted image though he does not know the original image content. And it is also hopeful that the original content should be recovered without any error after image decryption and message extraction at receiver side. A content owner encrypts the original image using an encryption key, and a data-hider can embed additional data into the encrypted image using a data-hiding key though he does not know the original content.

2. Data Extraction:

We will consider the three cases that a receiver has only the data-hiding key, only the encryption key, and both the data-hiding and encryption keys, respectively. With an encrypted image containing embedded data, if the receiver has only the data-hiding key, he may first obtain the values of the parameters from the LSB of the selected encrypted pixels. Then, the receiver permutes and divides the other pixels into groups and extracts the embedded bits from the LSB planes of each group. When having the total extracted bits, the receiver can divide them into original LSB of selected encrypted pixels and additional bits. Note that because of the pseudo-random pixel

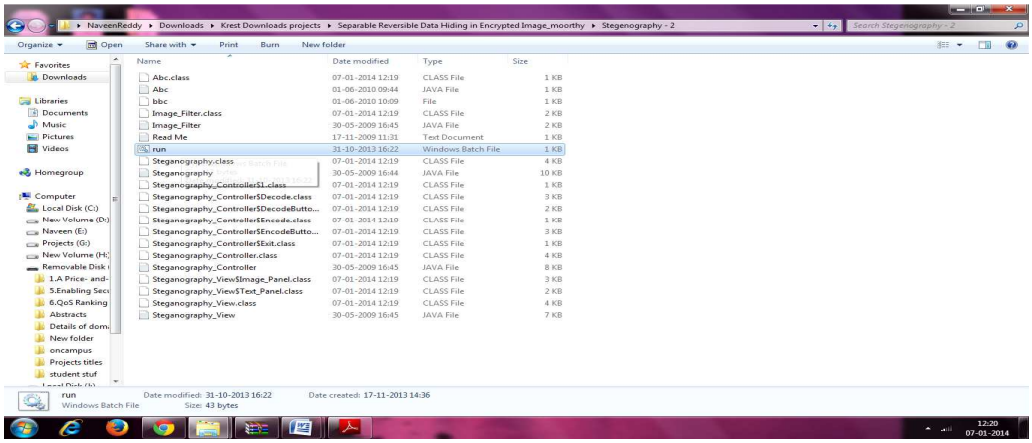
selection and permutation, any attacker without the data-hiding key cannot obtain the parameter values and the pixel-groups, therefore cannot extract the embedded data. Furthermore, although the receiver having the data-hiding key can successfully extract the embedded data, he cannot get any information about the original image content.

3.* Image Recovery*:

In this phase, we will consider the three cases that a receiver has only the data-hiding key, only the encryption key, and both the data-hiding and encryption keys, respectively. Note that because of the pseudo-random pixel selection and permutation, any attacker without the data-hiding key cannot obtain the parameter values and the pixel-groups, therefore cannot extract the embedded data. Furthermore, although the receiver having the data-hiding key can successfully extract the embedded data, he cannot get any information about the original image content.

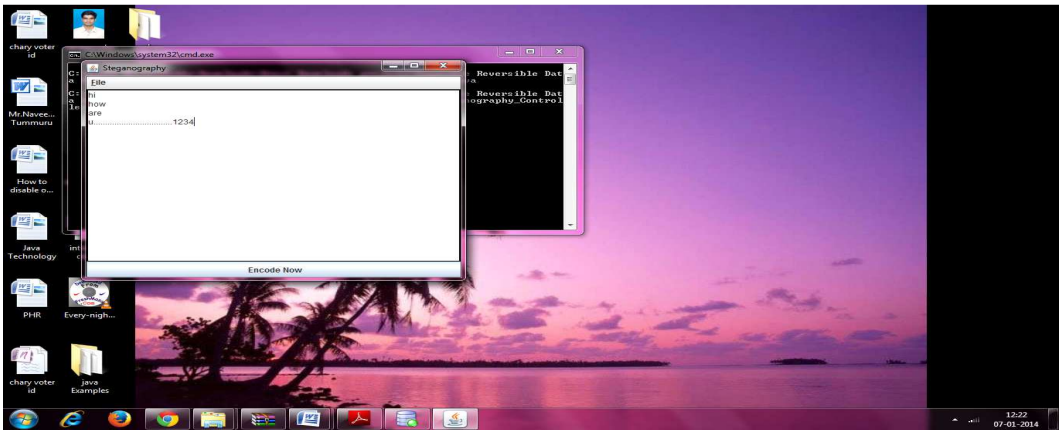
OUTPUT SCREENS:

Double click on run.bat



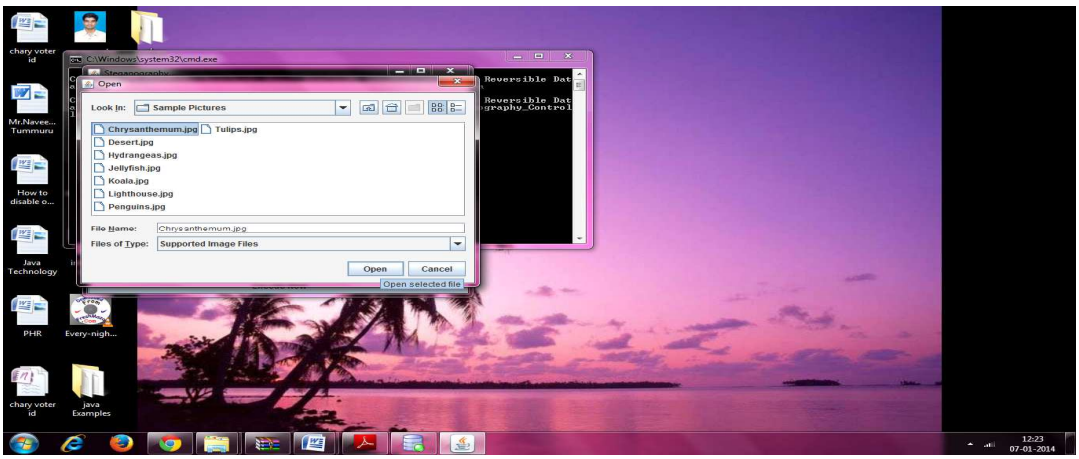
Home screen

Here we need to enter the text that to be hided in encrypted image format:

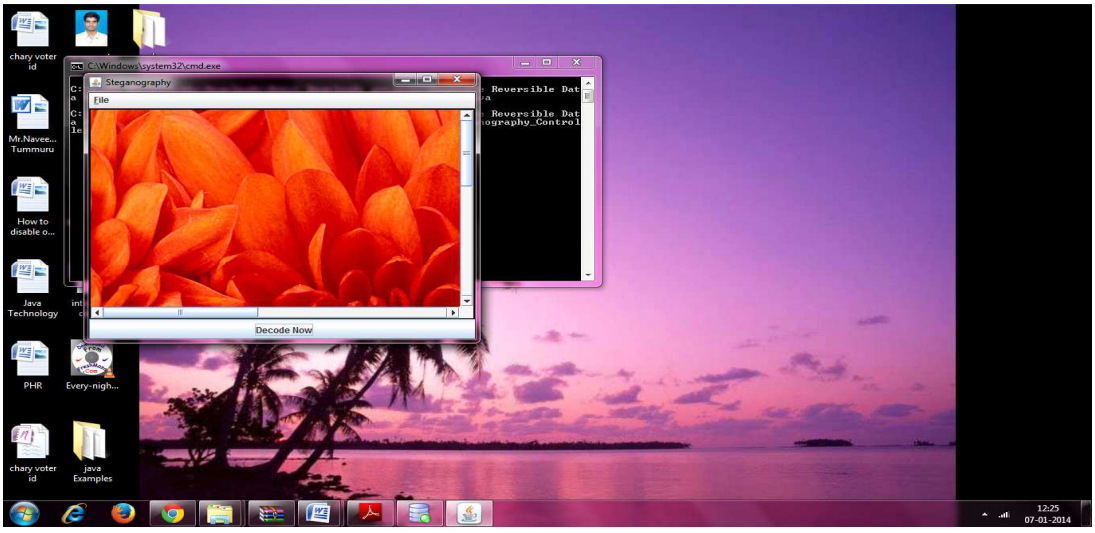


Entering the text click on Encode Now, it will ask us to select an image in which we have to hide the data:

Select any image, then give any out put file name:



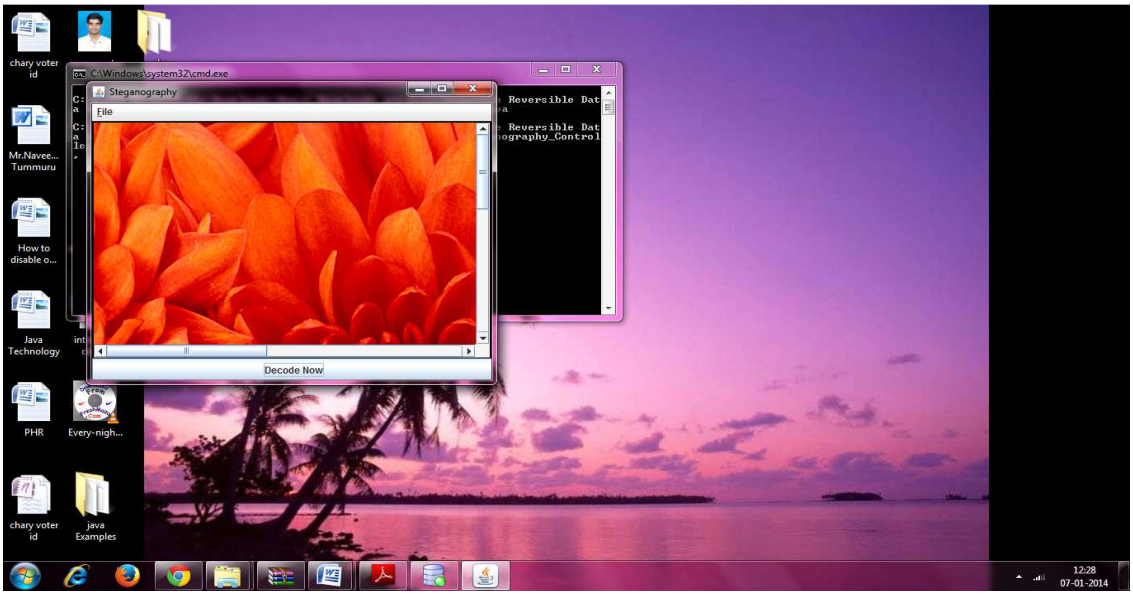
After the encryption:



After the encryption we need to decrypt the data hidden behind the output file:

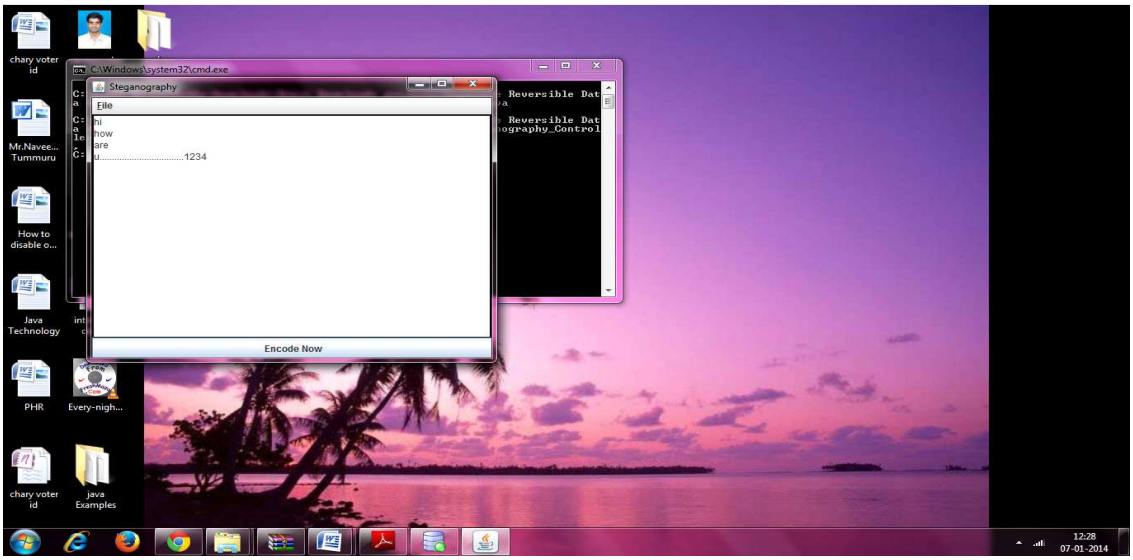
Then it will ask us to select a file that we have to decrypt:

Select the file:



Then click on decode now button:

Click on Ok, then it will show the hidden data:



CONCLUSIONS:

In this paper, a novel scheme for separable reversible data hiding in encrypted image is proposed, which consists of image encryption, data embedding and data-extraction/image-recovery phases. In the first phase, the content owner encrypts the original uncompressed image using an encryption key. Although a data-hider does not know the original content, he can compress the least significant bits of the encrypted image using a data-hiding key

to create a sparse space to accommodate the additional data. With an encrypted image containing additional data, the receiver may extract the additional data using only the data-hiding key, or obtain an image similar to the original one using only the encryption key. When the receiver has both of the keys, he can extract the additional data and recover the original content without any error by exploiting the spatial correlation in natural image if the amount of additional data is not too large. If the lossless compression method in or is used for the encrypted image containing embedded data, the additional data can be still extracted and the original content can be also recovered since the lossless compression does not change the content of the encrypted image containing embedded data. However, the lossy compression method in compatible with encrypted images generated by pixel permutation is not suitable here since the encryption is performed by bit-XOR operation. In the future, a comprehensive combination of image encryption and data hiding compatible with lossy compression deserves further investigation.

BIBILIOGRAPHY:

Analysis & Design of Information Systems - SennAdvanced Java Programming - Dietel and Dietel

Mastering JAVA 2 - John Zukowski

Java Server Programming - Apress

Software Engineering - Roger S Pressman