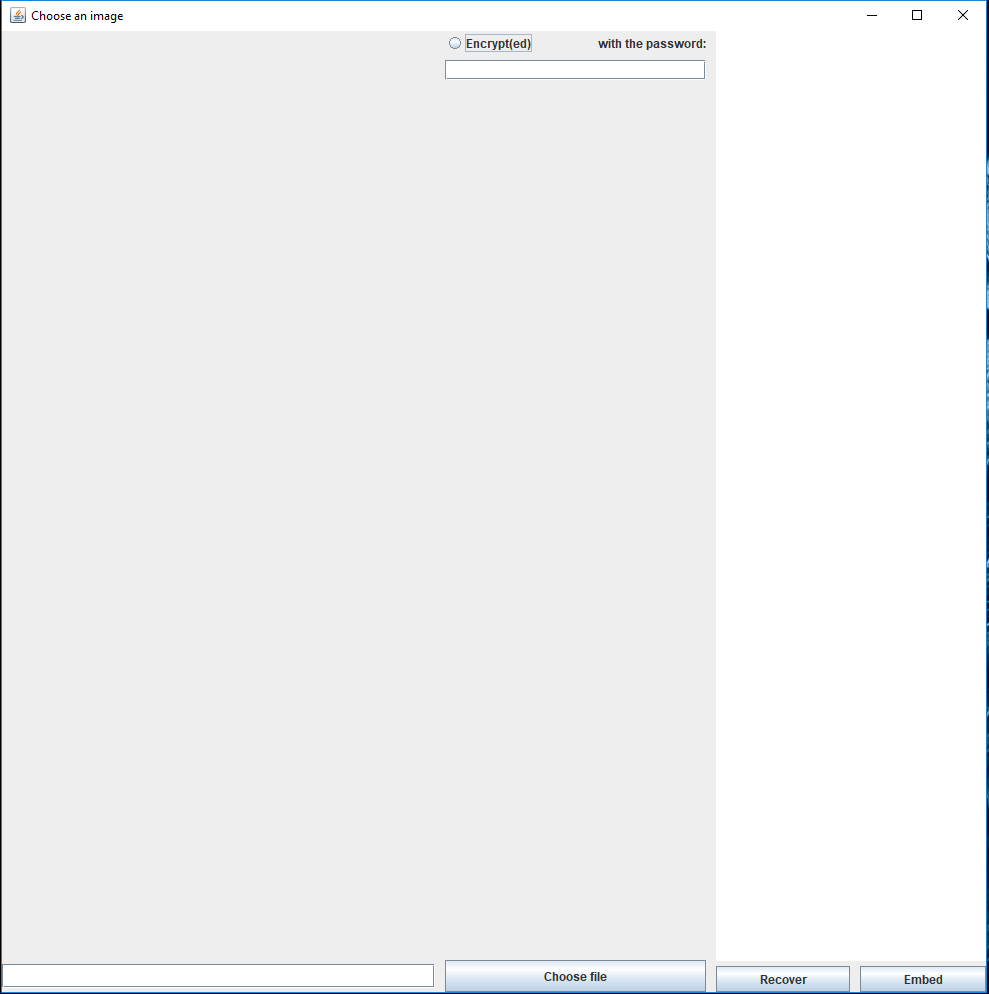
Steganography - Documentation

1. Introduction

Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video. The advantage of steganography over cryptography alone is that the intended secret message does not attract attention to itself. Plainly visible encrypted messages, no matter how unbreakable they are, arouse interest. In this project, a message is concealed in an image file. The user has the option to also encrypt the message with a password.

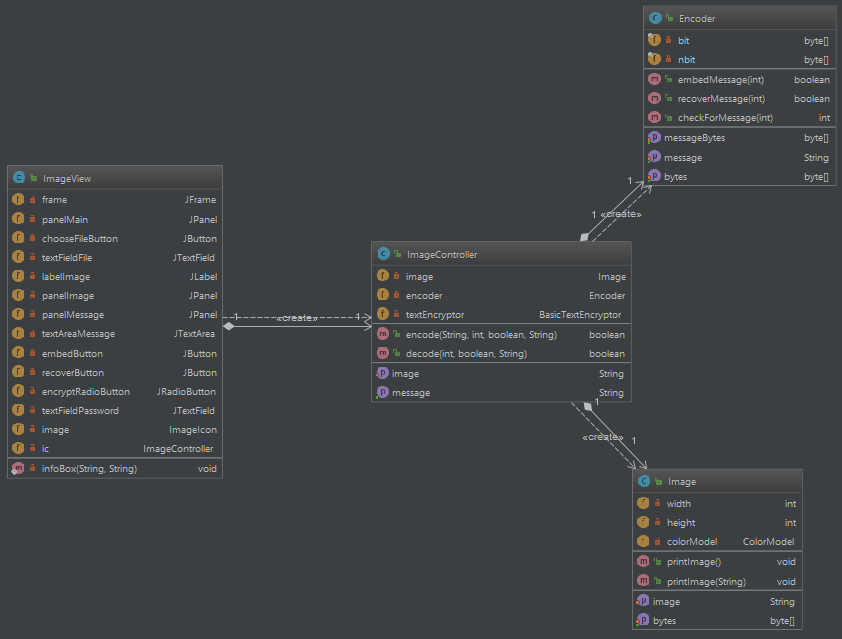
More information about steganography can be found [here](https://en.wikipedia.org/wiki/Steganography), or [here](https://www.youtube.com/watch?v=TWEXCYQKyDc).

1. Use cases



The application has a main window with three main elements. First, the user chooses an image file (jpeg, bmp, png) by pressing the “Choose file” button. The path to the file should appear in the text box next to it. The user can also manually introduce a path in said text box. A preview of the image should appear in the left part of the windows. After that, the user can choose to encrypt the message with a password, by checking the “Encrypt(ed)” radio button. If this option is chosen, the message can only be recovered with that password. The message to be embedded in the image is written in the rightmost text field. The message can contain white characters, and any UTF-8 character. To modify the image and include the message in it, the user needs to press the “Embed” button. A new file called “default.bmp” will be created in the directory of the executable. If the user presses the “Recover” button, the program will check for a message in the image. If there is no message present, the text “There is no message embedded in this file” will appear in the text field, otherwise the message will be shown.

1. Design and Classes



The Encoder class is used to modify the last n bits of each byte from a byte array. It needs to have an array of bytes and a message to be embedded. The number of bits to use can also be specified, but an even number is recommended. It can also recover a message from an array of bytes. The message that is embedded will start with the string “(embedded)” and the length of the message with 5 digits.

The Image class handles the files. It can read a file and get the RGB values in form of a byte array. It is also used to print an image based on a byte array.

ImageView handles the GUI of the application. Based on the input of the user it calls methods from the controller class and displays the output of the operation. This class only has event listeners and no data processing is done here.

ImageController is used to bind together all the other classes. It takes the path from the GUI and specifies it to the Image class so that it can extract the bytes. The controller takes the bytes and inserts them in the encoder. It also provides the encoder the message from the user. It handles the encryption/decryption of the message and provides the decoded message to the GUI.

1. Implementation details

The encoder class represents the bulk of the project. It works with two arrays of bytes: one that represents the image, and one that represents the message. The algorithm is as follows:

1. Take the byte from the message bytes
2. Take the first byte of the image bytes
3. Based on the number of bits used to encode, set the first cursor on the image bytes on the right position (n-1, where n is the number of bits), and the second cursor on the message bytes on the most significant bit
4. Change the bit on the selected positions and decrement both cursors
5. Repeat step 4 until one of the cursors is 0
6. If the message cursor gets to 0, check if there are more bytes, if there are get the next one, reset the cursor and repeat from step 4, else stop. If the image cursor gets to 0, get a new byte and rest the cursor.

The bits are changed based on a bitmap and the logic OR and AND operations.

The project uses an external library for encryption: [Jasypt](http://www.jasypt.org/) which can be downloaded [here](https://sourceforge.net/projects/jasypt/files/).

1. Missing features and further improvements

The application does not currently support encoding a different media file (such as audio or video), and it can’t save files as JPG/JPEG because the compression algorithm destroys the message.

The application does not support encoding another file in the image file.

In the future these features will be implemented alongside cosine transform coefficient modification in order to better hide the information.