Analysis and Implementation of Image Steganography Methods

A Major Project Report

Submitted in partial fulfillment of the

requirements for the award of the degree

of

Bachelor of Technology

in

INFORMATION TECHNOLOGY

Submitted

BY

VARUN KUMAR (Roll No. 1130328)

KULDEEP KUMAR (Roll No. 1130828)

PARMOD KUMAR (Roll No. 1130644)

DEPARTMENT OF COMPUTER ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

KURUKSHETRA-136119, HARYANA (INDIA)

April, 2017

CERTIFICATE

We hereby certify that the work which is being presented in B.Tech Major Project report entitled **"Analysis and Implementation of Image Steganography Methods"**, in partial fulfillment of the requirements for the award of the **Bachelor of Technology in Information Technology** is an authentic record of our own work carried out during a period from December 2016 to April 2017 under the supervision of **Dr. Mantosh Biswas**, Computer Engineering Department.

The matter presented in this project report has not been submitted for the award of any other degree elsewhere.

VARUN KUMAR (Roll No. 1130328)

KULDEEP KUMAR (Roll No. 1130828)

PARMOD KUMAR (Roll No. 1130644)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Mantosh Biswas

Asst. Professor, Computer Engineering

NIT Kurukshetra

Date:

List of Figures

List of Abbreviations

LSB: least Significant Bit

PNG: Portable Network Graphic

BMP: Bitmap image

JPEG: Joint Photographic Experts Group

GIF: Graphics Interchange Format

GUI: Graphical User Interface

IDE: Integrated Development Environment

ABSTRACT

Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the Internet. For hiding secret information in images, there exists a large variety of steganographic techniques some are more complex than others and all of them have respective strong and weak points. Different applications have different requirements of the steganography technique used. For example, some applications may require absolute invisibility of the secret information, while others require a larger secret message to be hidden.

In this project entitled "Analysis and Implementation of Image Steganography methods", we have focused on least significant bit modification techniques of spatial domain. We have analyzed and compared the variants of this algorithm in terms of effectiveness and hiding capacity. Effectiveness of the algorithm is measured by user evaluations determining at which point alteration to the images became apparent. We have developed an open source tool "Image-Stegano" <https://github.com/varunon9/Image-Stegano> in Java implementing the following steganographic methods-

- 1 bit, 2 bits, 3 bits and 4 bits LSB (hiding as well as extraction)
- 24 bits plane (Red, Green, Blue) and 32 bits plane (Red, Green, Blue, Alpha) analysis of image
- Bitwise XOR Implementation between LSB and payload
- Steganography Based on File Format

To make the "Image-Stegano" tool more effective we have also provided the users following options-

- Analysis of image using inversion and different color maps
- Grayscale analysis of image
- Altering threshold of image (Histogram)
- Providing metadata about the image
- Chunks analysis of .PNG images
- Extraction of appended Data (.PNG and .BMP files)

1. INTRODUCTION

1.1 What is Steganography?

"Steganography is the art and science of communicating in a way which hides the existence of the communication. In contrast to Cryptography, where the enemy is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by the cover message with the embedded cryptosystem. The goal of Steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second message present"[1]. In image steganography the information is hidden exclusively in images.

The idea and practice of hiding information has a long history. In Histories the Greek historian Herodotus writes of a nobleman, Histaeus, who needed to communicate with his son-in-law in Greece. He shaved the head of one of his most trusted slaves and tattooed the message onto the slave's scalp. When the slave's hair grew back the slave was dispatched with the hidden message [2]. In the Second World War the Microdot technique was developed by the Germans. Information, especially photographs, was reduced in size until it was the size of a typed period. Extremely difficult to detect, a normal cover message was sent over an insecure channel with one of the periods on the paper containing hidden information [3]. Today steganography is mostly used on computers with digital data being the carriers and networks being the high speed delivery channels.

2. LITERATURE SURVEY

2015: G. Prashanti and K. Sandhyarani have done survey on recent achievements of LSB based image steganography. In this survey authors discuss the improvements that enhance the steganographic results such as high robustness, high embedding capacity and un-detectability of hidden information. Along with this survey two new techniques are also proposed. First technique is used to embed data or secret messages into the cover image and in the second technique a secret gray scale image is embedded into another gray scale image. These techniques use four state table that produce pseudo random numbers. This is used for embedding the secret information. These two methods have greater security because secret information is hidden on random selected locations of LSBs of the image with the help of pseudo random numbers

generated by the table.

2014-2015: Savita Goel proposed a new method of embedding secret messages in cover image using LSB method using different progressions. Authors compare the quality of stego image with respect to cover image using number of image quality parameters such as Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), histograms and CPU time, Structure Similarity (SSIM) index and Feature Similarity Index Measure (FSIM). Their study and experimental results shows that their proposed method is fast and highly efficient as compared to basic LSB methods.

2015: Bingwen Feng, Wei Lu, and Wei Sun in their paper "Secure Binary Image Steganography Based on Minimizing the Distortion on the Texture" purposed a state of-the-art approach of binary image steganography. This technique is proposed to minimize the distortion on the texture. In this method of steganography firstly the rotation, complement and mirroring invariant texture patterns are extracted from the binary image. They also proposed a measurement and based on this proposed measurement this approach is practically implemented. Practical results show that proposed steganographic approach has high statistical security with high stego image quality and high embedding capacity.

2014: On the based on Huffman Coding, Amitava Nag present a novel steganographic technique of LSB substitution. Their technique basically focuses on high security, larger embedding capacity and acceptable level of stego image quality. Firstly Huffman tree is produced to encode every 8 bits of secret image. After encoding, they divide the encoded bits into four parts and have 0 to 3 decimal values. Location of embedding a message in cover image is determined by these decimal values. Experimental results show that it is very difficult for attacker to extract the secret information because Huffman table decrease the size of the cover image. Proposed techniques just have acceptable level of PSNR values and lie between 30 dB to 31dB.

2012: In this author proposes an enhanced LSB algorithm for image steganography. In this proposed work they only embed secret information in blue component of the RGB colour space. In their technique first M×N size cover image is selected. After selection of cover image only blue component is used for embedding secret information. They also make use of pixel filters to access the best regions to embed information in cover image to obtain 5 best possible rate.

Experimental results show that this technique reduces the distortion level of cover image and stego image has very good visible quality and changes in cover image are negligent to Human Visual System (HVS). This method reduces the leap in colour scale because only blue components are used to embed the secret information.

2010: On the bases of Human visual system (HVS) X. Qing proposed a new technique in which sensitive information is embedded in all planes of RGB components of an image. In this technique multiple plan-bit is used with adaptive nature of information hiding algorithm. This proposed method has high embedding capacity than traditional LSB method and low computational complexity. Proposed system also has good quality of stego image.

2009: H. Yang presented a new adaptive LSB based method for image steganography. It uses the pixel adjustment technique for better stego image quality. This adaptive LSB substitution results in high hidden capacity. LSB based image steganography method is proposed. To hide the data common bit pattern is used. According to the message and the pattern bits LSB's of pixels are modified. This method has low hidden capacity.

3. STEGANOGRAPHY TECHNIQUES

3.1 LSB

Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image [4]. The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the red, green and blue colour components can be used, since they are each represented by a byte. In other words, one can store 3 bits in each pixel. An 800 × 600 pixel image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data.

3.2 BIT PLANE

Instead of highlighting gray level images, highlighting the contribution made to total image appearance by specific bits might be desired. Suppose that each pixel in an image is represented by 8 bits. Imagine the image is composed of 8, 1-bit planes ranging from bit plane 1-0 (LSB) to bit plane 7 (MSB).In terms of 8-bits bytes, plane 0 contains all lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all high order bits. Separating a digital

image into its bit planes is useful for analyzing the relative importance played by each bit of the image, implying, it determines the adequacy of numbers of bits used to quantize each pixel, useful for image compression.

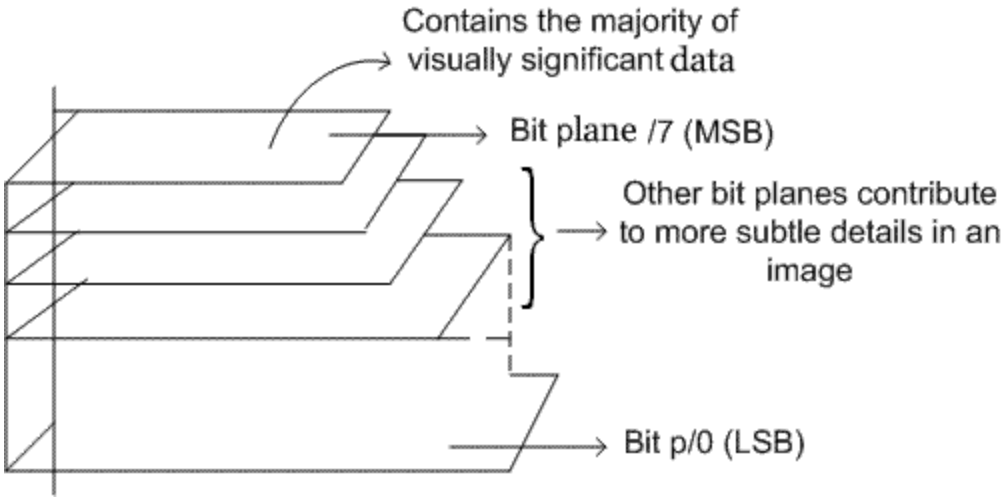

Figure 1: Bit Plane

3.3 SPIRAL EMBEDDING

The Spiral Embedding orders the pixels of the image in a spiral pattern to prevent the embedding from being as easily decoded. The Spiral Embedding has two main ideas that allow it to be decoded successfully and help thwart visual attacks. The first is that metadata about the image's contents are embedded into known locations. This information makes it possible to decode the stego object and retrieve the secret message. The second is that the data is serialized and embedded in a pattern that destroys the ability to decode the message in a visual attack.The Spiral Embedding begins by building a vector containing all the data that will be embedded into the cover including the metadata and the message contents. The dimensions of the message are embedded losslessly into the first 32 positions in the vector as unsigned 16-bit integers. A bit representing the vector’s content is written into the LSB of the cover in a spiral pattern from the outside in, pixel by pixel.Decoding a stego object created with the Spiral Embedding is simply a process of reading in the stored dimensions and then following the same spiral pattern that

governed the embedding. The values of the LSBs of the stego object are read into a vector. When the embedded data has been read in its entirety, the vector is partitioned to create a new image with the message's original dimensions. The result of this embedding can then be displayed.

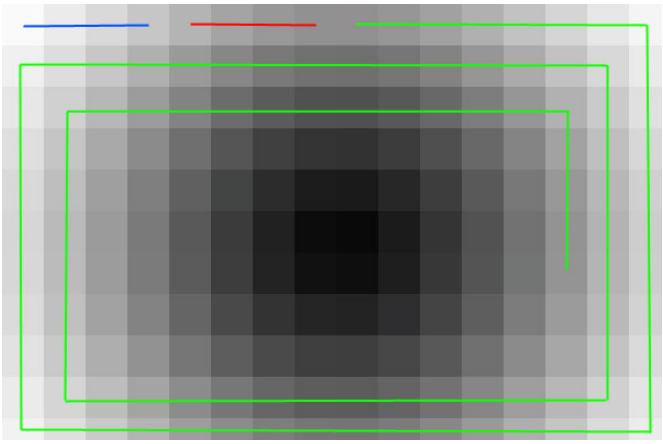

Figure 2 : The Spiral Embedding pattern. The height information is embedded into the blue line pixels, the width information into the red pixels, and the message into the green pixels.

3.4 METADATA MANIPULATION

Metadata is basically data about data. If we think of a image file as our data, metadata would include information such as the name of the image, the title, dimensions , width , height, Video and audio files contain the same types of data.

The Exchangeable Image File (Exif) format is a standard used for recording information about image and sound files created with a digital camera. There are many data fields that can be used to hide information. If we right click on a saved .jpg file and select properties, we will see a small subset of those fields.

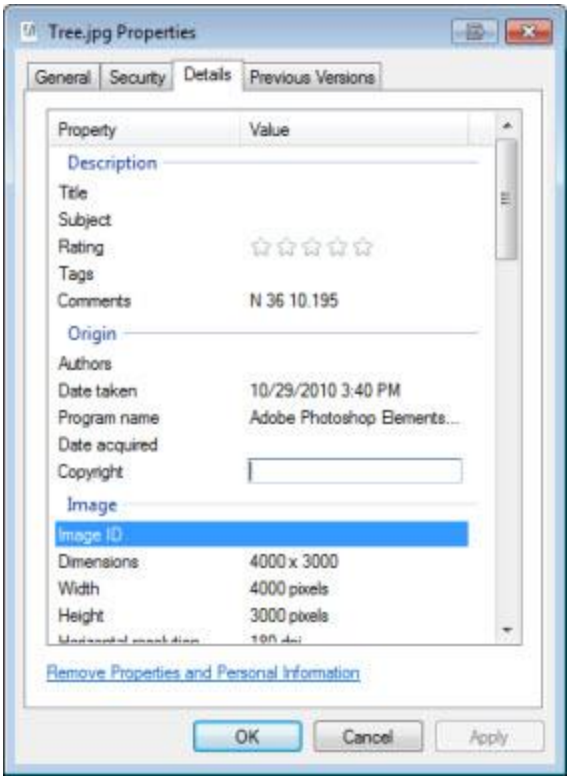

Figure 3: Meta Data Using File Properties Windows

3.5 ALTERING THRESHOLD

Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images. The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity is less than some fixed constant T or a white pixel if the image intensity is greater than that constant.

3.6 APPENDED DATA

A simple and common steganography technique is to append data at the end of image file. This technique works because most image viewers ignore any appended data and hence a stego message remains hidden. Linux ‘cat’ command can be used for appending data at the end of image file.

3.7 PNG CHUNK ANALYSIS

A PNG file starts with an 8-byte signature. After the header comes a series of chunks, each of which conveys certain information about the image. Chunks declare themselves as critical or ancillary, and a program encountering an ancillary chunk that it does not understand can safely ignore it. This forms the basis of steganography. Data can be embedded in unknown ancillary chunks and it will be ignored by image viewers or decoders.

A chunk consists of four parts: length (4 bytes, big-endian), chunk type/name (4 bytes), chunk data (length bytes) and CRC (cyclic redundancy code/checksum; 4 bytes). The CRC is a network-byte-order CRC-32 computed over the chunk type and chunk data, but not the length.

Chunk types are given a four-letter case sensitive ASCII type/name; compare FourCC. The case of the different letters in the name (bit 5 of the numeric value of the character) is a bit field that provides the decoder with some information on the nature of chunks it does not recognize.

The case of the first letter indicates whether the chunk is critical or not. If the first letter is uppercase, the chunk is critical; if not, the chunk is ancillary. Critical chunks contain information that is necessary to read the file. If a decoder encounters a critical chunk it does not recognize, it must abort reading the file or supply the user with an appropriate warning. The case of the second letter indicates whether the chunk is "public" (either in the specification or the registry of special-purpose public chunks) or "private" (not standardised). Uppercase is public and lowercase is private. This ensures that public and private chunk names can never conflict with each other (although two private chunk names could conflict). The third letter must be uppercase to conform to the PNG specification. It is reserved for future expansion. Decoders should treat a chunk with a lowercase third letter the same as any other unrecognised chunk. The case of the fourth letter indicates whether the chunk is safe to copy by editors that do not recognize it. If lowercase, the chunk may be safely copied regardless of the extent of modifications to the file. If uppercase, it may only be copied if the modifications have not touched any critical chunks[6].

3.8 CHANGING COLOUR MAP

In computing, indexed color is a technique to manage digital images' colors in a limited fashion, in order to save computer memory and file storage, while speeding up display refresh and file

transfers. It is a form of vector quantization compression.

When an image is encoded in this way, color information is not directly carried by the image pixel data, but is stored in a separate piece of data called a palette: an array of color elements. Every element in the array represents a color, indexed by its position within the array. The individual entries are sometimes known as color registers. The image pixels do not contain the full specification of its color, but only its index in the palette. This technique is sometimes referred as pseudocolor or indirect color, as colors are addressed indirectly. This technique can be exploited to hide data in image using suitable palette or colour map. To decode data we must try different random colour maps.

4. PROJECT DESCRIPTION

4.1 PROJECT STATUS

The following features have been implemented in Image-Stegano Tool-

A. Analysis of image in different bit planes
B. Analysis of Indexed Images using different colour tables
C. Hiding image in different Bit Planes
D. Analysis of image by altering threshold
E. Hiding text data using LSB and Spiral Embedding
F. Extracting metadata of images
G. Data chunks analysis of PNG images
H. Extracting appended data for BMP and PNG images

4.2 SCREENSHOTS

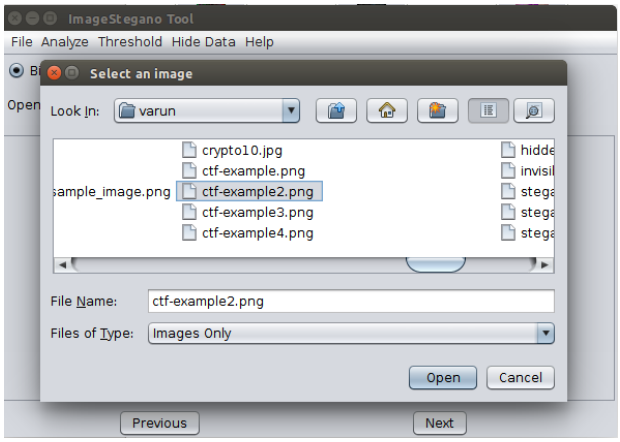

Figure 4: Opening an image to analyse

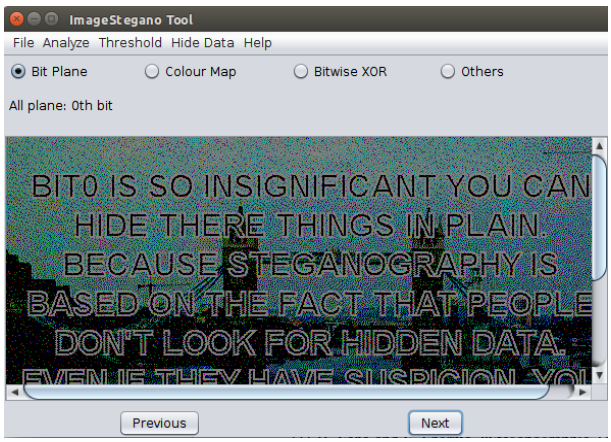

Figure 5: Hidden Image found in 0th bit plane

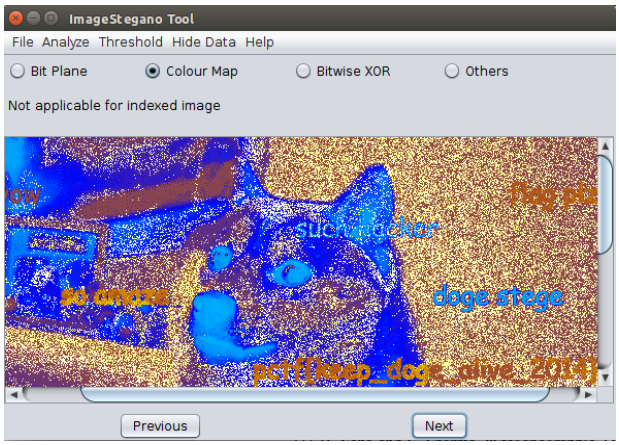

Figure 6: Changing color map

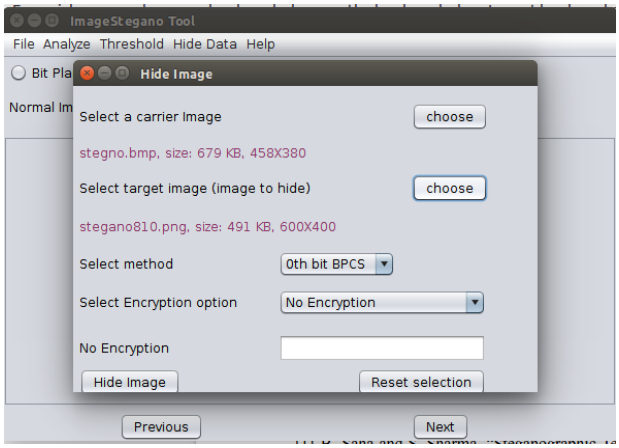

Figure 7: Hide Image GUI

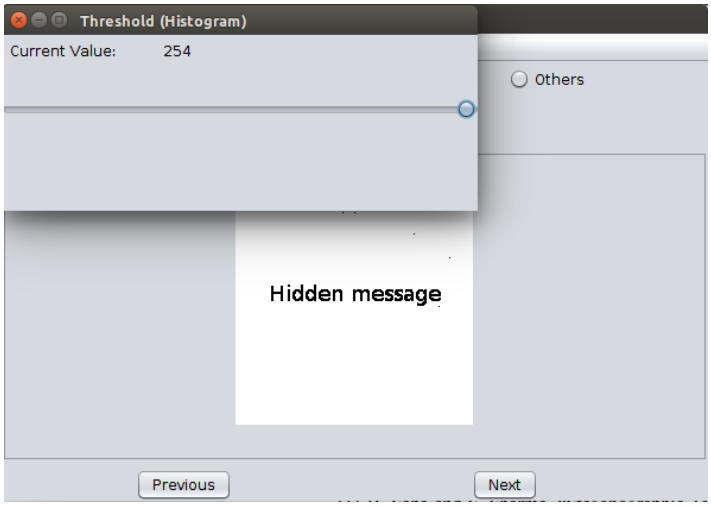

Figure 8: Hidden data visible on altering threshold

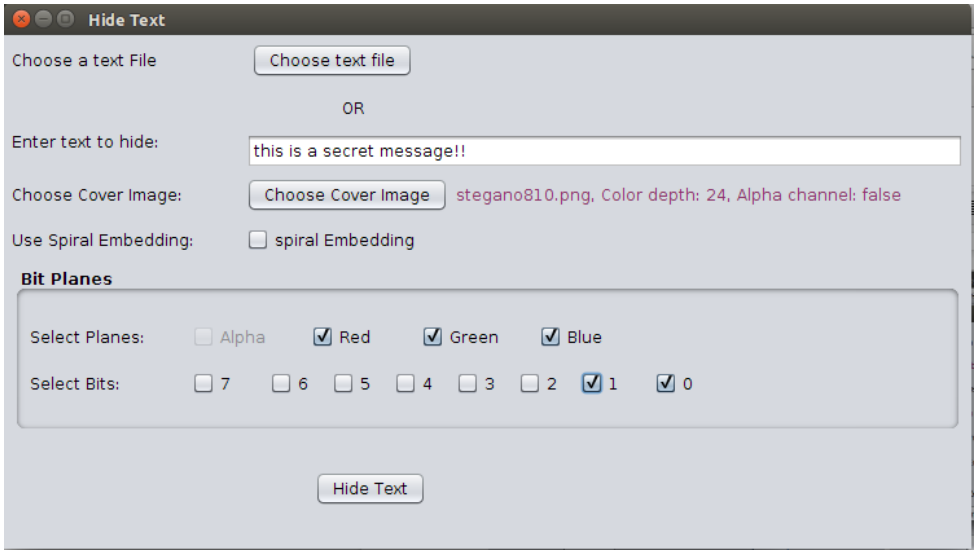

Figure 9: Hide Text using LSB

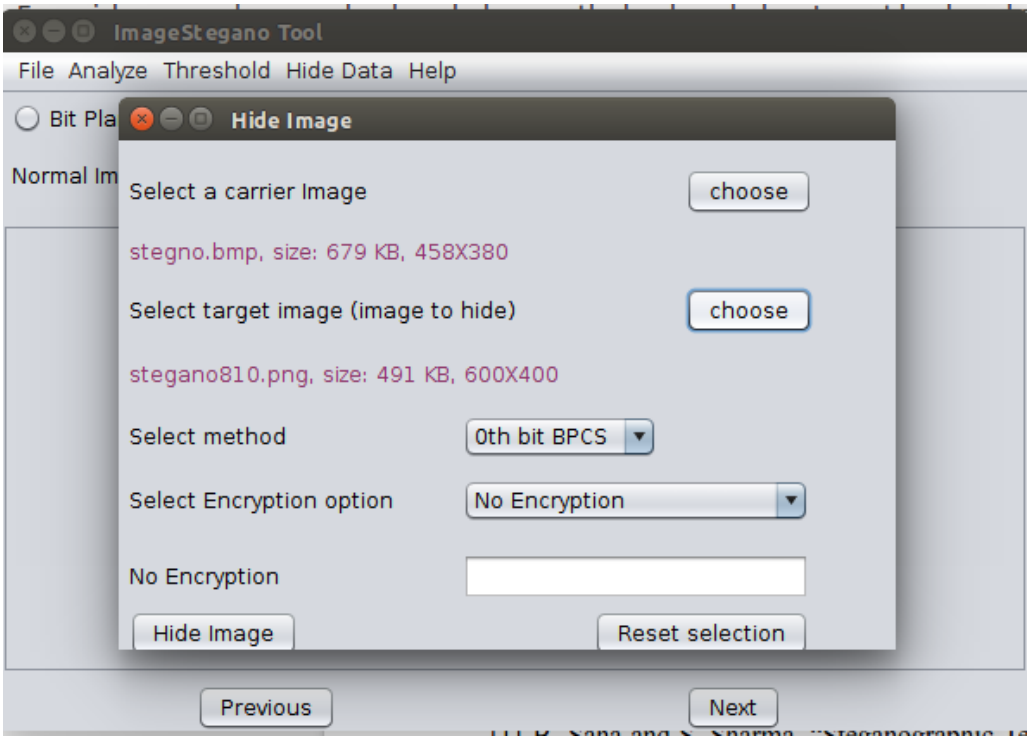

Figure 10: Hide Image inside bit plane of cover image

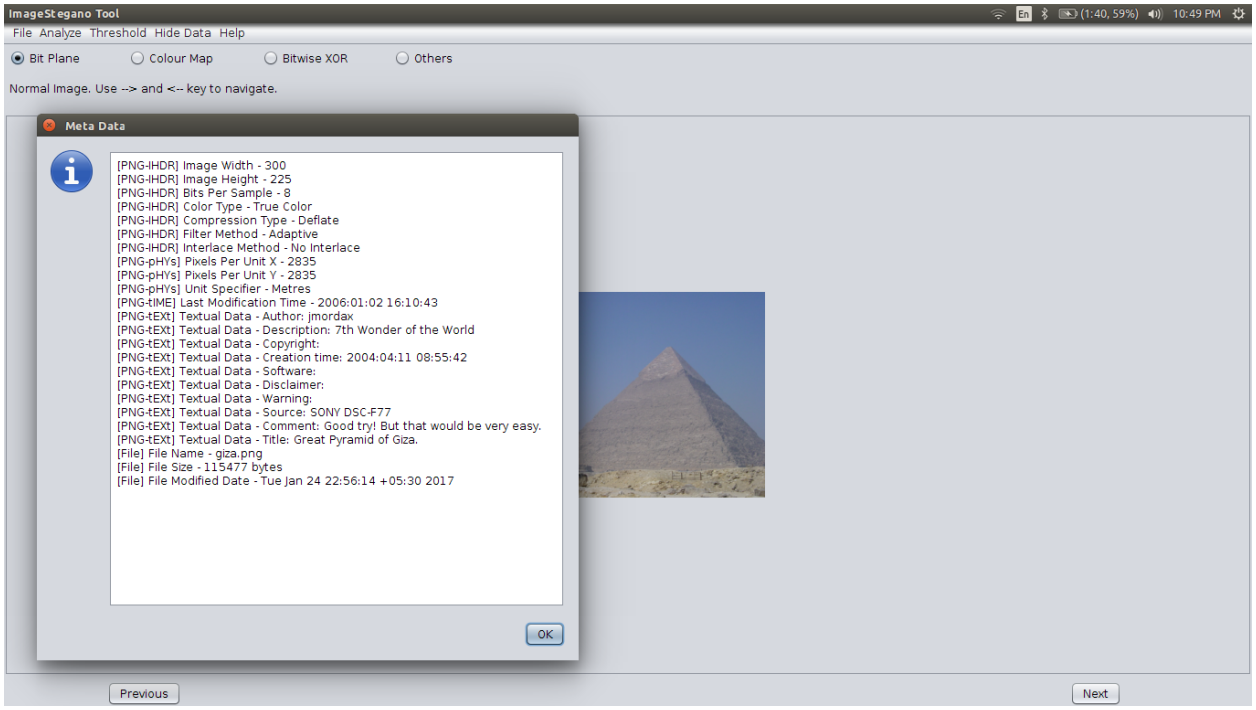

Figure 11: Extracting metadata from image

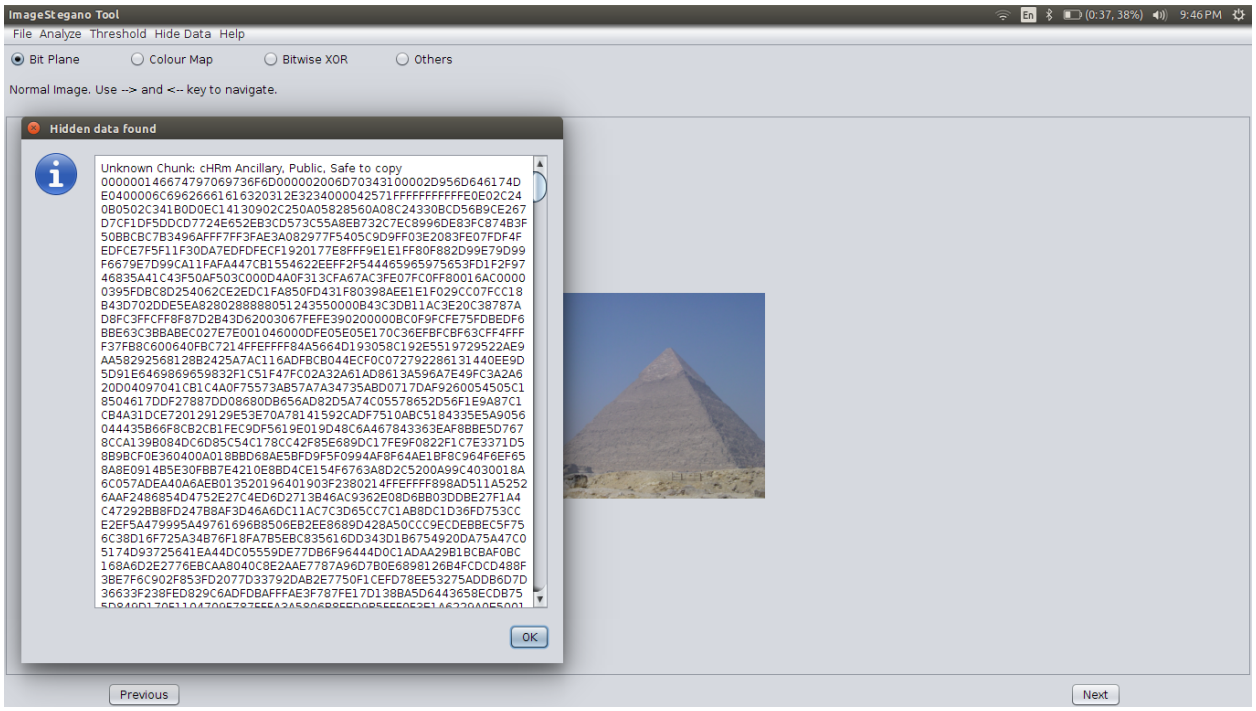

Figure 12: PNG Chunk Analysis

4.3 DEPENDENCIES

We have developed Image-Stegano tool in Java using Netbeans IDE. It uses https://github.com/drewnoakes/metadata-extractor library for metadata extraction.

4.4 FLOW CHART

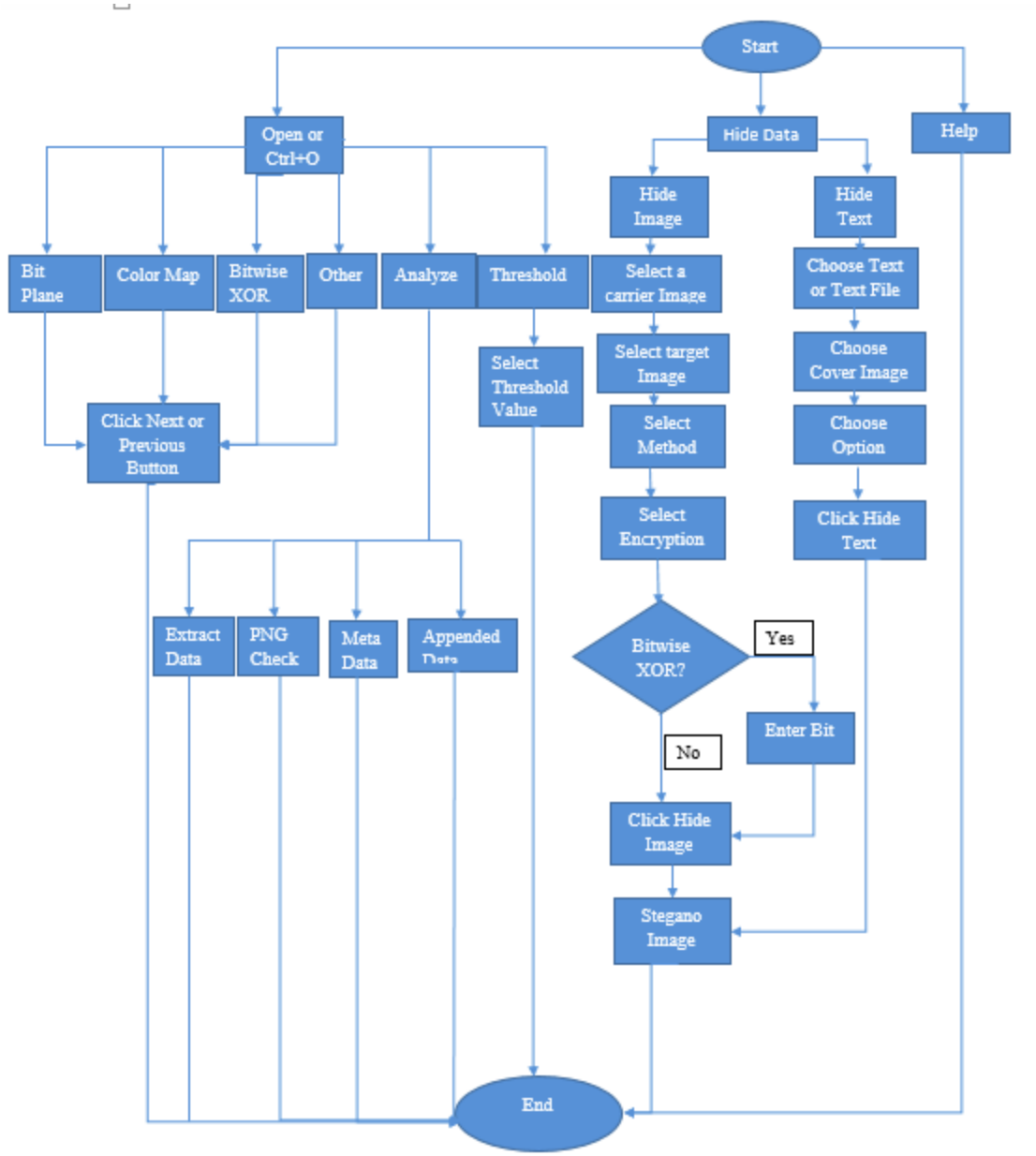

Figure 13: Flow Chart

4.4 LIMITATIONS

Image-Stegano tool has following limitations-

A. All the steganography methods are of spatial domain i.e. directly modified image pixels.
B. Appended data extraction does not work for JPEG images.
C. Tool does not provide options for compression or encryption of payload.
D. JPEG images cannot be used as cover image to hide data.

5. RESULTS

5.1 Comparing hiding image in different bit planes

Figure 14: Cover Image for hiding secret image

Figure 15: Secret image hidden in 0th bit plane

Figure 16: Secret image hidden in 1st bit plane

Figure 17: Secret image hidden in 2nd bit plane

Figure 18: Secret image hidden in 3rd bit plane

Figure 19: Original secret image

5.2 Analyzing different bit planes for any hidden image

Figure 20: Original RGB colour image

Figure 21: 7th bit plane

Figure 22: 6th bit plane

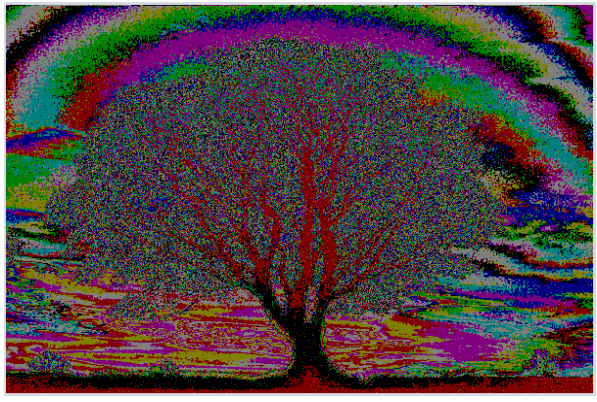

Figure 23: 4th bit plane

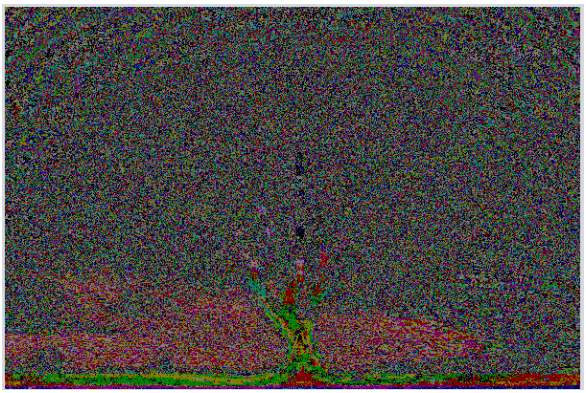

Figure 24: 2nd bit plane

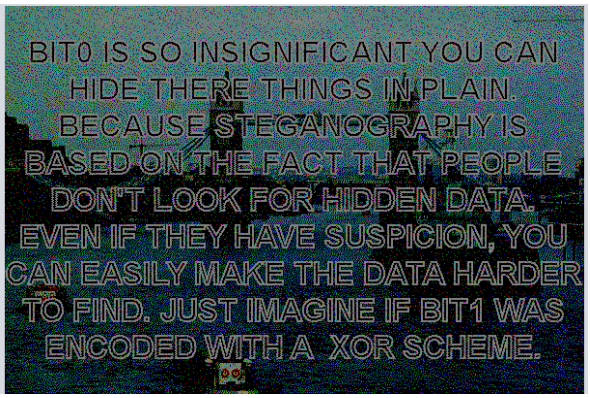

Figure 25: Hidden image in 0th bit plane

5.3 Changing colour map of indexed image to see any hidden data

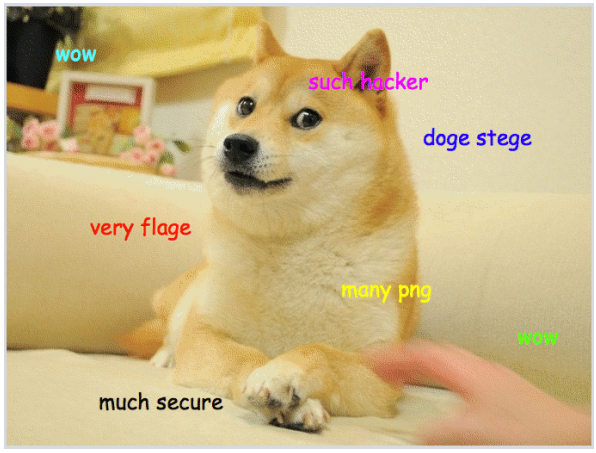

Figure 26: Original Indexed image

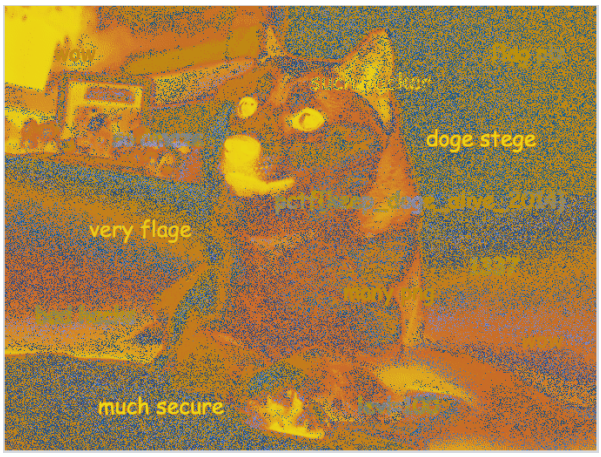

Figure 27: Colour Map 1

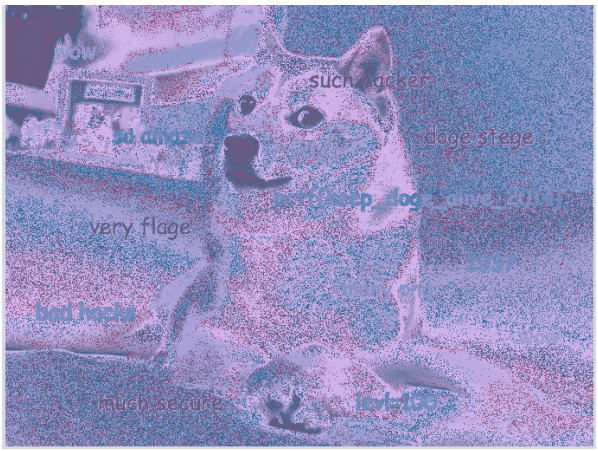

Figure 28: Colour Map 2

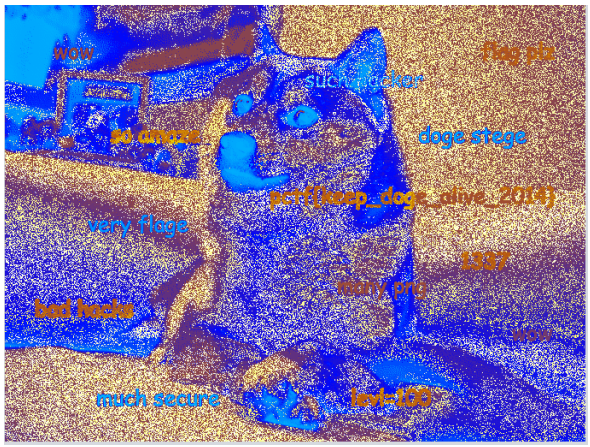

Figure 29: Colour Map 3

5.4 Hiding Text in different bits

Figure 30: Original 24 bit PNG image

Figure 31: Data hidden in 0th bit

Figure 32: Data hidden in 0th and 1st bits

Figure 33: Data hidden in 3rd and 4th bits

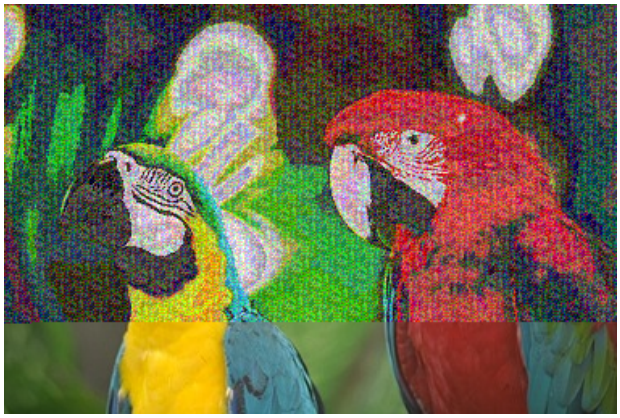

Figure 34: Data hidden in 6th bit

Figure 35: Data hidden in 7th bit

6. COMPLETE CONTRIBUTORY SOURCE CODE

Complete source code is available at https://github.com/varunon9/Image-Stegano

7. CONCLUSION

Image-Stegano tool can be used by individuals/ organization. This tool is platform independent and is handy for analyzing common steganography methods. Though there are several others steganography tools available but Image-Stegano can be said as combination of many tools and since it is open source so new functionalities will be added over time by volunteer contributors. Besides functional requirements, we have also focused on non-functional requirements (Nice UI and UX, Efficient). The tool has limitation that it does not accept jpeg images as carrier image to hide data. Also this tool does not use compression and encryption of payload.

REFERENCES

[1] B. Saha and S. Sharma, "Steganographic Techniques of Data Hiding using Digital Images", in Defence Science Journal, vol. 62, no. 1, 2012 January, pp. 11-18.

[2] Silman, J., "Steganography and Steganalysis: An Overview", SANS Institute, 2001.

[3] Jamil, T., "Steganography: The art of hiding information in plain sight", IEEE Potentials, 18:01, 1999.

[4] Johnson, N.F. & Jajodia, S., "Exploring Steganography: Seeing the Unseen", Computer Journal, February 1998.

[5] Sei-ichiro Kamata, et al: Depth-First Coding for Multi-Valued Pictures Using Bit-Plane Decomposition, IEEE Trans. on CT, Vol.43, No.5, pp.1961-1969, May, 1995.

[6] https://www.w3.org/TR/PNG-Chunks.html