DETECTION OF PRESENCE MARKS IN STEGOCONTENT

SIA CHIA MENG

THESIS REPORT SUBMITTED IN FULFILMENT OF THE DEGREE OF COMPUTER SCIENCE (COMPUTER SYSTEM AND NETWORKING

DETECTION OF PRESENCE MARKS IN STEGOCONTENT

SIA CHIA MENG CA10008

THESIS SUBMITTED IN FULFILMENT OF THE DEGREE OF COMPUTER SCIENCE

FACULTY OF COMPUTER SYSTEM AND SOFTWARE ENGINEERING

2013

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS

JUDUL: …………………………………………………………………………….

SESI PENGAJIAN: …………………………………

SAYA ……………………………………………………………….(HURUF BESAR)

Mengaku membenarkan tesis/laporan PSM ini disimpan di Perpustakaan Universiti Malaysia Pahang dengan syarat-syarat kegunaan seperti berikut:

1. Tesis/Laporan adalah hakmilik Universiti Malaysia Pahang.
2. Perpustakaan Universiti Malaysia Pahang dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institut pengajian tinggi.
4. **Sila tandakan (√)

□ SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972) *

□ TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan) *

□ TIDAK TERHAD

Disahkan Oleh

………………………………. ………………………………………

Alamat tetap: Penyelia:

Tarikh:……………………… Tarikh:……………………………..

*Sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh tesis/laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

Date: 27 May 2013

Student Name : Sia Chia Meng

Student Number: CA10008

Signature: ________________________________

SUPERVISOR DECLARATION

I hereby declare that I have read this thesis and in my opinion this thesis/report is sufficient in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Computer System and Networking).

Signature :………………………………………

Supervisor Name:

Date :……………………………………………

ACKNOWLEDGMENTS

Firstly, I feel lucky to have persevere, patience and good health throughout the duration of this final year project research. I am very fortunate to have Madam Wan Nurulsafawati binti Wan Manan as my research supervisor. Also, I would like to express my high appreciation to my supervisor Madam Wan Nurulsafawati binti Wan Manan. The supervision and support that she gave truly help the progression and smoothness of the progress of my final year project. The co-operation is much indeed appreciated.

I would like to show my appreciation to Dr. Rahmah binti Mokthar, the coordinator of final year project for giving guidance, help, support and useful information to me for completion of my report. I also want to thanks the lecturers and staffs of Faculty of Computer System and Software Engineering for their cooperation during I complete the final year project that had given valuable information, suggestions and guidance in the preparation this final year project report.

Moreover, deepest thanks and appreciation to my parents and family members for their encouragement and full of support for the report completion, from the beginning till the end. Also thanks to all of my friends, they have been contributed by supporting my work and help myself during the final year project progress.

Last but not least, my thanks to my Personal Advisor, Mr. Mohamed Ariff bin Ameeden, for great commitment and cooperation during my final year project.

ABSTRACT

Steganography is an art of hiding message by embedding message within other types of media for example text, audio or image files. The main purpose of Steganography is attempts to hide the existence of communication. For steganography, the users totally do not want to let anyone know they are sending messages. The objectives of this project are study current steganography tools and the methods used for detection of presence marks in stegocontent, develop steganalysis tools which implement with graphical user interface (GUI), and verify and detect the marks in stegocontent in image files to assist investigation process. Peak Signal-to-Noise Ratio (PSNR) steganalysis is chose as the method used in this application software because it is use for detect various Least Significant Bits (LSB) modification techniques. LSB is the method that most commonly used in steganography to hide message.

ABSTRAK

Steganografi adalah seni menyembunyikan mesej dengan menerapkan mesej di dalam lain-lain jenis media contohnya teks, audio atau imej. Tujuan utama steganografi adalah untuk menyembunyikan kewujudan komunikasi. Bagi steganografi, pengguna benar-benar tidak menginginkan orang lain tahu bahawa mereka menghantar mesej. Objektif projek ini adalah mengaji alat-alat steganografi semasa dan kaedah yang digunakan untuk mengesan tanda-tanda kehadiran di stegocontent, membangunkan alat steganalysis yang melaksanakan dengan Graphical User Interface (GUI), dan mengesahkan atau mengesan tanda-tanda dalam stegocontent dalam fail imej untuk membantu proses penyiasatan . Peak Signal-to-Noise Ratio (PSNR) steganalysis adalah memilih sebagai kaedah yang digunakan dalam aplikasi perisian ini kerana ia digunakan untuk mengesan pelbagai Least Significant Bits (LSB) teknik pengubahsuaian. LSB adalah kaedah yang paling biasa digunakan dalam steganografi untuk menyembunyikan mesej.

TABLE OF CONTENT

LIST OF TABLE

LIST OF FIGURES

LIST OF ABBREVIATIONS

BMP	Bitmap format
DCT	Discrete Cosine Transform
DFD	Data Flow Diagram
DWT	Discrete Wavelet Transform
GIF	Graphics Interchange Format
GUI	Graphical User Interface
JPEG	Joint Photographic Experts Group
LSB	Least Significant Bits
MSE	Mean Square Error
PNG	Portable Network Graphics
PoV	Pairs of Value
PSNR	Peak Signal-to-Noise Ratio
QIM	Quantization Index Modulation
RQP	Raw Quick Pairs
RS	Regular and Singular groups

CHAPTER 1:

INTRODUCTION

1.1 Introduction

Steganography is an art of hiding message by embedding message within other types of media for example text, audio or image files. According to Kumar & Pojar (2010), steganography is not same as cryptography. Steganography is about secrecy and cryptography is concern on privacy. The cryptography is used to protect the information from reveal to public. For steganography, the users totally do not want to let anyone know they are sending messages. Few centuries ago, the first uses of steganography by Demaratus, the king of Sparta to send a warning about a forthcoming attack to Greece by writing the message on the wood and cover the message with wax upon to hide the secret message. There is also the ancient people hiding the message by tattooed the message or map on shaved head of a person. The message is then become hidden when the hair is grown and they will shaved the head again when they want to see the message. Those are some of the techniques or method that ancient people use to hiding the message. Now, steganography become more popular due to internet and multimedia as the internet makes the message transfer fast and free. Nowadays, steganography can be used for water marking where a message is hidden in the “carrier” so that its sources can be track or verify (Curran & Devitt, 2008). However, some of the users use steganography technique to commit crime because it is hard to detect. There is some of the steganalytic software used to detect the media that have hidden message but it is not fully implement with GUI. This makes new users hard to learn to get familiar to the functions and operations of the software as they have insufficient technical knowledge. Besides that, there are limited numbers of steganalytic software in the market. In conclusion, although the use steganography technique is not intend to doing something bad, there are cases that show that some people misuse it to commit crime. So, this steganalytic software can help the investigation on stegocontent to prevent crime. It also make the user can use it easily and convenient.

1.2 Problem Statement

The basic structure of Steganography is made up of three components: the “carrier”, the message, and the key. The main purpose of Steganography is attempts to hide the existence of communication. According to Cole (2003), three principles can be used to measure the effectiveness of a steganography technique. Those principles are amount of data, difficultly of detection and difficulty of removal. For amount of data, explanation will be the more data can be hided, more great the technique. Difficulty of detection relates to how easy it is for someone to detect that a message has been hidden. Once the amount of data hidden increases in a file, the risk the message be detected also becomes higher. Difficulty of removal suggests that the hidden data cannot be able to remove easily. Actually steganography can be used to protect intellectual property from digital robbery (copyrights) by injecting the copyright marks and serial number in electronic medium such as books and audio. However, there are some irresponsible users abused steganography techniques to commit crimes. For example, criminals can communicate with each other secretly by hiding the messages using steganography techniques to commit crimes like selling and transfer drugs and avoid getting caught by polices. It is very difficult to investigate those crimes since there are too many techniques to hide the message. Besides that, there are limited numbers of software or program that help to detect the marks of stegocontent. Furthermore, most of the steganography analysis software are lack of Graphical User Interface (GUI) so many new users does not understand how to operate them and it is complex to use.

1.3 Objectives

This project comprehends the following objectives:

i. To study current steganography tools and the methods used for detection of presence marks in stegocontent.
ii. To develop steganalysis tools which implement with graphical user interface (GUI).
iii. To verify and detect the marks in stegocontent in image files to assist investigation process.

1.4 Scope

i. This steganalysis application software targets all computer users.
ii. All of the computer users can use this tool to detect stegocontent for precaution or security purposes.
iii. This steganalysis tool is focus on the detection of stegocontent in image files (JPEG/JPG, PNG) only.
iv. This steganalysis tool detects only Least Significant Bits (LSB) modification in image files.

1.5 Thesis Organization

This project report includes chapter 1, chapter 2, and chapter 3. Chapter 1 provides a broad overview of the thesis. In chapter 1, it includes introduction, problem statements, objectives, scopes and thesis organization. Chapter 2 reviews the previous research of steganography and comparison of steganography and steganalysis methods. Chapter 3 describes the approach used to develop the application. It includes prototype interface, use case diagram, flow chart of the application software.

Chapter 2:

Literature Review

In recent years the growth in the quantity of available Steganography tools on Internet can be seen obviously. In short, Steganography is the art or technique to hide messages within types of media. The goal of steganography is to keep the existence of a message hidden or to hide the fact that communication is taking place. In contrast, the goal of cryptography is to make the messages cannot be understood (Druid, 2006). Nowadays, Internet is widely use to transfer and store information or messages. This means that Internet can be considered to be a storehouse of steganographic materials (Callinan & Kemick, 2006). It is believed that criminals and terrorist organizations may be communicating secretly through the use of steganography. Therefore, Steganalysis, the techniques of detecting hidden messages using Steganography is necessary to detect hidden data. However, some steganographic techniques are particularly difficult to detect without the original sources (Johnson & Jajodia, 1998).

2.1 Introduction of Image Steganography

According to Curran & Devitt(2008), digital images are the most widely used medium for steganography today and it take advantage of our limited visual perception of colour. The most popular image formats on the internet are the graphics interchange format (GIF), joint photographic experts group (JPEG) format, the portable network graphics

(PNG) format and the bitmap format (BMP). Queirolo(2006) states that large images are the most desirable for steganography because they have more space to hide the data. This field is expected to continually grow as fast development of computer graphics power and technology (Calpe,2006).

2.2 Image Steganography Methods

Image steganography has been widely studied by researchers. There are a variety of methods used in which information can be hidden in images. The following is the methods of image steganography:

2.2.1 Replacing Least Significant Bit

Least significant bit (LSB) insertion is a common and simple way to embedding information in a cover image (Johnson & Jajodia, 1998). For instance, a simple scheme proposed by Lee and Chen (2000), is to place the embedding data at the least significant bit (the 8th bit) of each pixel in the cover image. The modified image is called stego-image. Manchanda,Dave and Singh(2007) found that altering LSB doesn't change the quality of image to human perception but it is sensitive to image processing attacks like compression, cropping etc. In addition, Nagham Hamid, Abid Yahya, R. Badlishah Ahmad & Osamah M. Al-Qershi(2012) discovered that changes of cover image using LSB techniques are very difficult to be recognized by the human eye because they are being too small.

2.2.2 Replacing Moderate Significant Bit

Chan and Chang(2001) showed the use of the moderate significant bits(the 4th bit) of each pixel in the cover image to embed the secret message. This method improves sensitivity to modification, but it degrades the quality of stego-image.

2.2.3 Transformation Domain Techniques

It is another familiar data hiding techniques by use the transformation domain of digital media to hide information. Functions such as the discrete cosine transform (DCT) and the discrete wavelet transform (DWT) are widely applied in this technique. The DCT transforms a signal from an image representation into a frequency representation, by grouping the pixels into 8 × 8 pixel blocks and transforming the pixel blocks into 64 DCT coefficients each (Krenn, 2000). According to Nagham Hamid, Abid Yahya, R. Badlishah Ahmad & Osamah M. Al-Qershi(2012), wavelet transform clearly partitions the high-frequency and low-frequency information on a pixel by pixel basis therefore wavelet is used in image steganography. Manchanda,Dave and Singh(2007) stated that the messages are hided in the significant areas of the cover image by using these methods, which makes them robust against image processing attacks like compression and cropping.

2.3 Steganalysis Techniques

The goal of steganalysis is to identify suspected information streams, determine existence of any hidden messages and recover the hidden message if possible (Si, 2004). A steganalyst is trying to determine the existence of a hidden message instead of knowing which bits carry what information (Zhang & Ping, 2006). There are several forms are taken to attacks and analysis on hidden information for example detecting, extracting and disabling or destroying hidden information (Chandramouli & Memon, 2006). Curran and Devitt(2008) indicated that steganalysis can be classified into two categories which are Passive Steganalysis and Active Steganalysis. Passive Steganalysis only involves detection while Active Steganalysis process is complete only after the hidden data is removed, destroyed or strategically altered to render it useless. Provos and Honeyman(2006) concluded that the main purpose of steganography is failed once its existence is revealed by steganalysis although the secret content is not exposed.

2.3.1 Steganalysis attacks

There are three types of Steganalysis attacks:

2.3.1.1 Visual or Aural attacks

They consist of striping away the significant parts of a digital content in order to facilitate a human’s visual inspection for anomalies (Wayner, 2002). The idea of Visual attacks is to remove all parts of the image covering the message so human eye can now distinguish whether there is a hidden message or still image content (Westfeld and Pfitzmann, 1999). A common test is to show the LSBs of an image.

2.3.1.2 Structural attacks

Sometimes the format of the digital file changes as hidden information is embedded so these changes lead to an easily detectable pattern in the structure of the file format (Westfeld and Pfitzmann, 1999). Identifying those characteristic structure changes can detect the presence of hidden file, for example in palette based steganography the palette of image is changed before embedding data to reduce the number of colors so that the adjacent pixel color difference should be very less as the result, this shows groups of pixels in a palette have the same color which is not the case in normal images (Bennett, 2004).

2.3.1.3 Statistical attacks

These types of attacks are more effective and successful as they reveal the smallest alterations in an images statistical behavior (Bhattacharyya & Sanyal, 2012). Statistical tests can reveal modified image by determining an image’s statistical properties deviate from the norm (Provos & Honeyman, 2006). Digital pictures of natural scenes have distinct statistical behavior. Mercuri(2004) discovered that we can determine whether or not an image has been altered with proper statistical analysis, making forgeries mathematically detectable. Therefore, this Steganalysis method is to collect statistical evidences about the presence of hidden messages in images, and use them to verify the existence of hidden content on given images (Bishop, 2006).

2.3.1.4 Classification of attacks based on information available (Johnson, 2000)

- **Stego only attack**: Only stego object is available for analysis.
- **Known cover attack**: Both cover and stego are known.
- **Known message attack**: In some cases message is known. Analyzing the stego object pattern for this message embedded may help to attack similar systems.
- **Chosen stego attack**: Steganographic algorithm and stego object are known.
- **Chosen message attack**: Steganalyst implements many steganographic tools for a chosen message and analyses these stego objects with the one which is to be analyzed and try to find the algorithm used in these process.
- **Known stego attack**: Cover, object and the steganographic tool used are known.

	Stego object	Original cover object	Hidden message	Stego algorithm or tool
Stego only attack	x			
Known cover attack	x	x		
Known message attack	x		x	
Chosen stego attack	x			x
Chosen message attack	x			
Known stego attack	x	x		x

Table 2.1 Summary of attacks based on information available

2.3.2 Steganalytic Methods

Steganalytic Methods	**Description**	**Targeted Steganographic Techniques**
RS (Regular and Singular groups) steganalysis	Sensitivity of dual statistics based on spatial correlation of pixels to LSB randomization due to steganographic embedding is used in analysis.	Various LSB modification techniques
PoV(Pairs of Value)-based Chi-square test	Chi-square test checks whether the occurrence of each pair of values tends to become equal, indicating some data is embedded.	Steganography based on swapping pairs of values of pixel gray levels, colors, or DCT coefficients
Palette checking	Peculiarity in palette ordering is a clear sign of systematic modification.	Steganography in palette images
RQP (Raw Quick Pairs) method	Method based on analyzing the increased number of close-color pairs caused by embedding.	LSB embedding in true-color images
Check JPEG compatibility	Method detects unusual departure from the JPEG signature inherent in images initially stored in JPEG format.	Space-domain steganography using images initially

		stored in the JPEG format
Histogram analysis	Method reveals discreteness or periodicity in particular coefficients due to quantization-related modification.	QIM (Quantization Index Modulation) or other quantizationrelated embedding methods
Universal blind detection	Statistical quantities constructed using high-order statistics, and a detection model established with the threshold obtained in a training process.	Various steganographic techniques
Peak Signal-to-Noise Ratio (PSNR) analysis	Calculation of Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) values to identify concentration of the noise inside the stegoimage.	LSB modification technique.

Table 2.2 Comparison of Steganalytic methods (Generate from Wang and Wang, 2004)

2.4 Conclusion

Peak Signal-to-Noise Ratio (PSNR) steganalysis is chose as the method used in my program because it is use for detect various LSB modification techniques. LSB is the method that most commonly used in steganography to hide message. So, it is necessary to develop a program to detect it.

Chapter 3:

Methodology

3.1 Introduction

Methodology is an organized, documented set of procedures and guidelines for problem solving with components like techniques, tools, methods, and tasks. Methodology includes a diagramming process for documenting the outcomes of the procedures, approach for carrying out the procedure, and determining the quality of results of the procedures. The methodology may include publication research, questionnaires, interviews, surveys and other research techniques, and may include present and previous information. It describes the methods to be used. Research design, the population to be studied, and the research instruments or tools to be used are discussed in the methodology. In short, methodology is the way of how to conduct research.

3.2 Research Approach

Research approach is the method chosen for development of the application which includes describing the process used to develop software or system product. It describes the activities performed and how the development phases follow each other to ensure the success in the process of system development. Agile modeling is chosen to ensure the application development process runs smoothly.

3.2.1 Agile Modeling

Agile modeling is a framework or process that describes the activities performed at each stage of a software and system development project. Agile modeling methodology divides the software development process into several phases to make the whole process become more organize and easier to achieve, control, and manage thus help out software developer to develop an organized system. The main purpose of Agile modeling is to produce a high quality system that meets or exceeds customer requirements or expectations, complete works within time and cost estimates, works effectively and efficiently by utilizes current and planned Information Technology infrastructure, and cost-effective to maintain and enhance. Agile modeling ensuring that all functions, user requirements and agency strategic goals and objectives are met. Each phase of agile modeling continues and refines what is done in the previous phase. Commonly known development phases in agile modeling are planning, analysis, design, implementation, verification and maintenance.

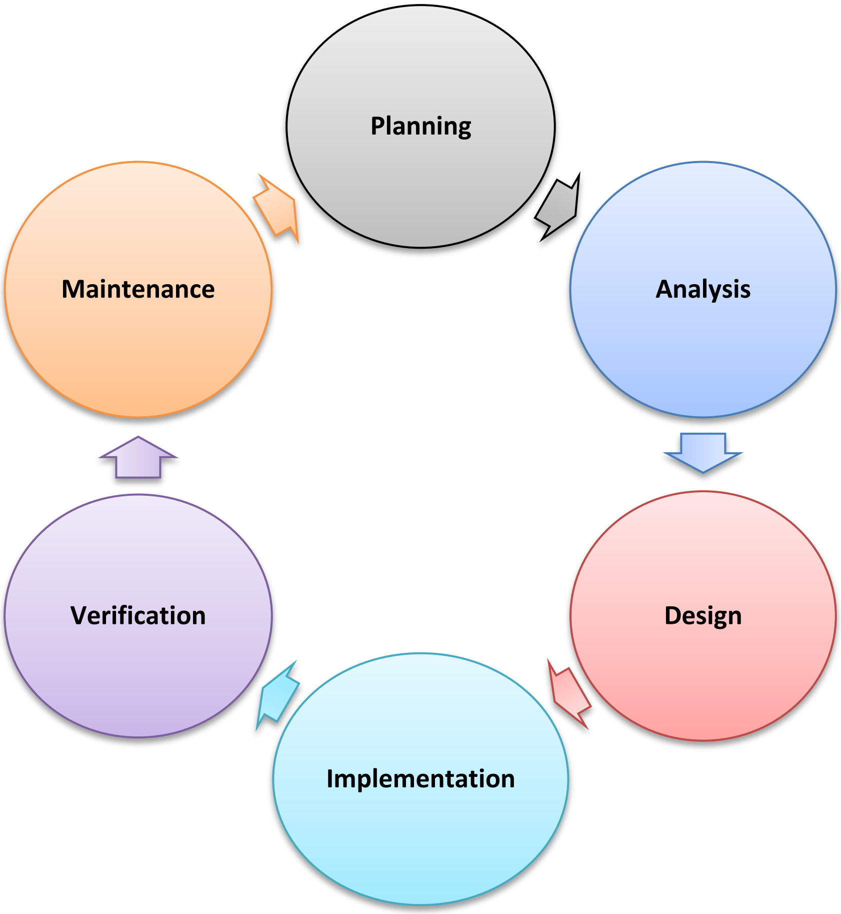


Figure 3.1 Phases in Agile Modeling

Agile modeling is chosen because it brings benefits to system development. Firstly, agile modeling can avoid unexpected high cost and low expected benefits. It allows the progress to be reviewed at the conclusion of each phase. Furthermore, agile modeling has detail steps and processes and well defined user input. For agile modeling, deadline date is more adhered because the completion time is fixed. The other advantages of agile modeling include evaluate costs and completion targets, development and design standards are good, relevant software and system documentation, maximize productivity, and improve quality of systems. Besides that, the system development process will become easy to monitor, manage and control as well for maintenance. Compare with waterfall methodology, agile modeling is more flexible and it allows backward process to change the decisions and works that had made in previous phase after each phase is completed and it is welcome for changing requirements, even late in development.

3.3 Implementation of the Project

Development phases in agile modeling are planning, analysis, design, implementation, verification and maintenance. The details and the description of each phase are important to developer to understand activities of each phase and the progress of the project.

3.3.1 Planning Phase

It is the process of understanding why the system should be built, determines the goals and objectives of the project, and defining its requirements. It also includes feasibility study from several different perspectives, technical, economic, and organization feasibility aspects. A project management plan and other planning documents are developed. Provide the basis of requiring the resources needed to achieve solution. The risks and various project-planning approaches are defined. The existing system is evaluated and deficiencies and weaknesses are identified during this phase. The examples of existing systems to be studied are StegDetect, Stegspy and Xsteg. Gantt chart is produced during this phase. All the tasks are planned and arranged. The timeline of the project is shown in Appendix A.

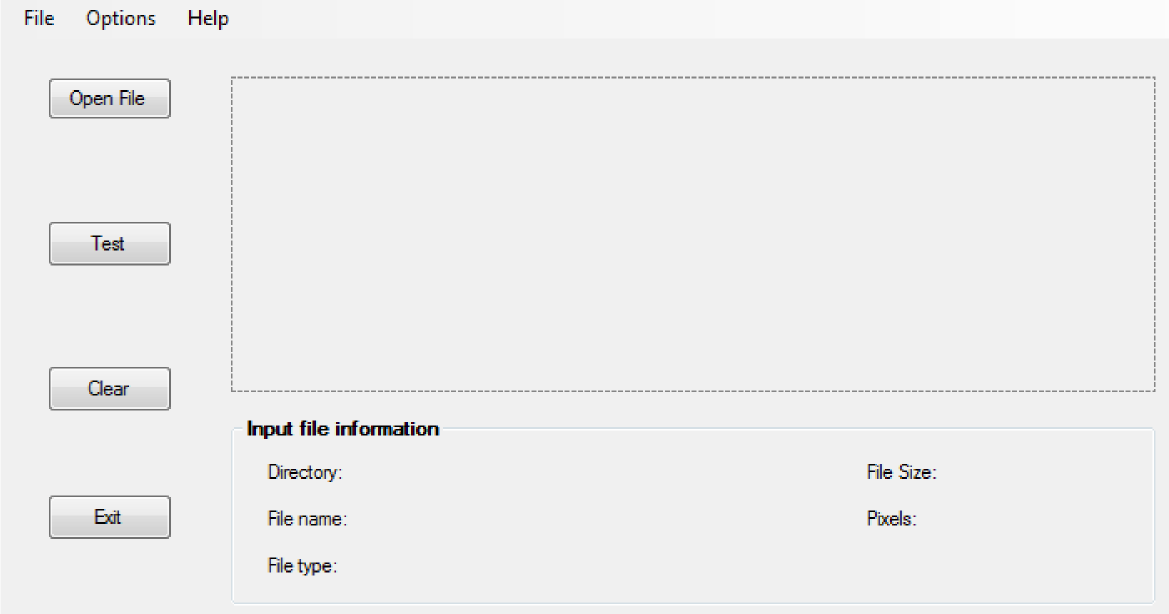
3.3.2 Analysis Phase

Information needs and requirements of the end users, project goals, organizational environment, and any system presently being used are analyzed and also develop the functional requirements of a system that can meet the needs of the users. Besides that, the requirements are recorded in a document. The requirements documentation should be referred to throughout the rest of the system development process to ensure the developing project meets and fulfill the user needs and requirements. Problems are identified and suggestions are recommended for improving the system functioning. For example, find out problems exist and attempt to fix the system. During this phases, the problems are found in existing systems are they do not have GUI and not suitable for new users to use as they does not have enough technical knowledge to operate it. Suitable technique of steganalysis and tools is identified during this phase. The PSNR technique is chosen to detect LSB modification in image files and tools suitable for develop a new application is MATLAB.

3.3.3 Design Phase

During this phase, all detail functional requirements are translated into preliminary and complete designs. Decisions are made to address how the system will meet functional requirements. A general system design emphasizes the functional features of the system. After that a final or detailed system design is produced which specifying all the technical detail needed to develop the system. There are several techniques used for describing the system design of the system for example flowchart and data flow diagram (DFD). The design describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudocode and other documentation. Prototype interface design and use case diagram are created during this phase. Use case diagram describes the relationship among the application, user and developer. GUI is implemented to the application so that new users can operate it more easily.

3.3.3.1 Prototype Interface Design



Input file information

Directory:

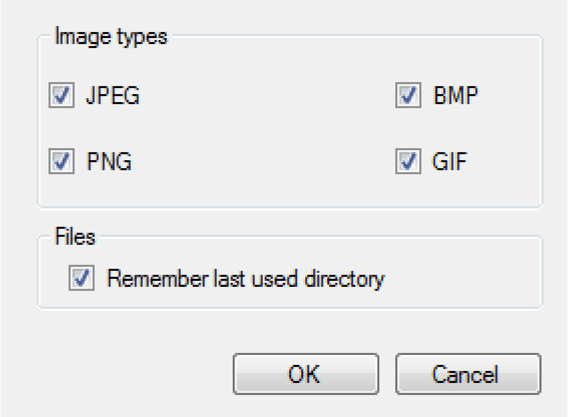
File Size:

File name:

Pixels:

File type:

Figure 3.2 Prototype interface design



☑ JPEG ☑ BMP

☑ PNG ☑ GIF

☑ Remember last used directory

Figure 3.3 Prototype option interface

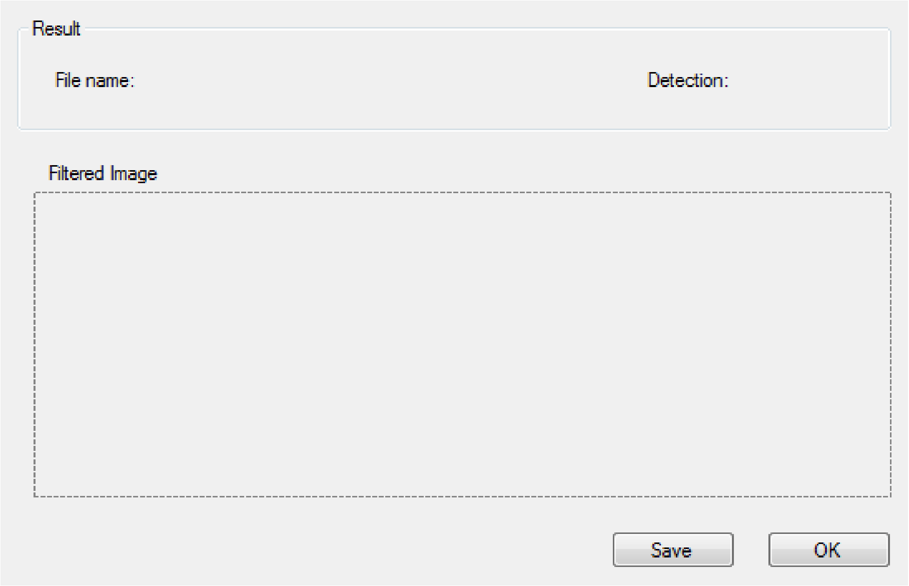


Figure 3.4 Prototype result interface

The “open file” button is used to choose an image file to test. The test button is used to test the chosen image. Click the test button the result interface will come out and show filtered image, file name and detection result. In the result interface, users can click save button to save the results. The clear button is used to clear all the information and chosen image. Exit button is used to exit the software. The input file information of chosen file like directory, file name, file size, and pixels is show in the interface. The image will show in the interface also. Click the “option” function, the option interface will pop out and the users can choose desirable option for image type or remember last used directory.

3.3.3.2 Flow Chart

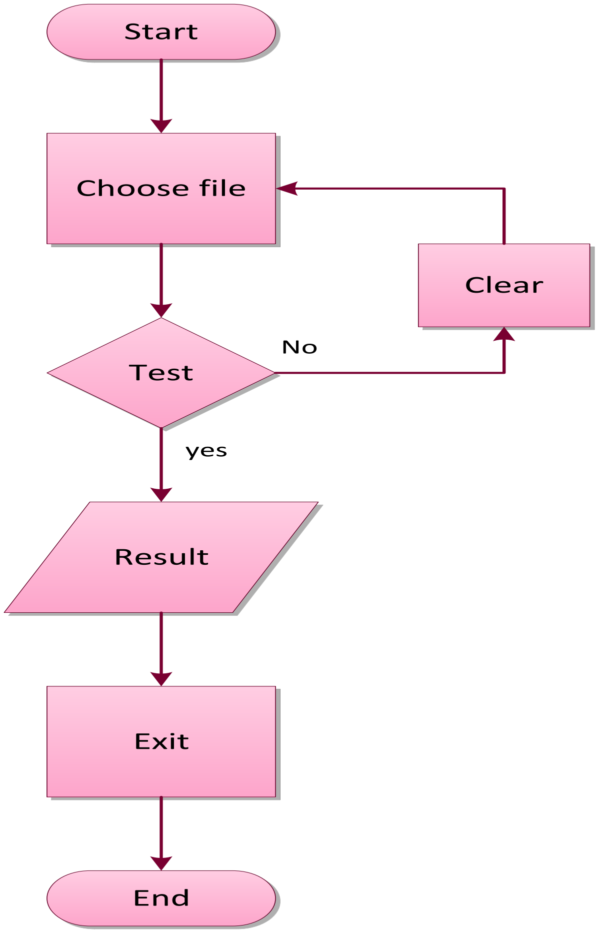


Figure 3.5 Flowchart of system

First, user start with execute the software. Then, choose an original image file (JPEG/JPG, PNG format) and suspected image to test. After image files are chose, choose the test function for testing and analyzing the image file. If the user chooses a wrong file, user can clear the chosen image and choose a new image file. After test, result will come out and show on the interface. Finally user can exit the software if not using anymore.

3.3.3.3 Use Case Diagram

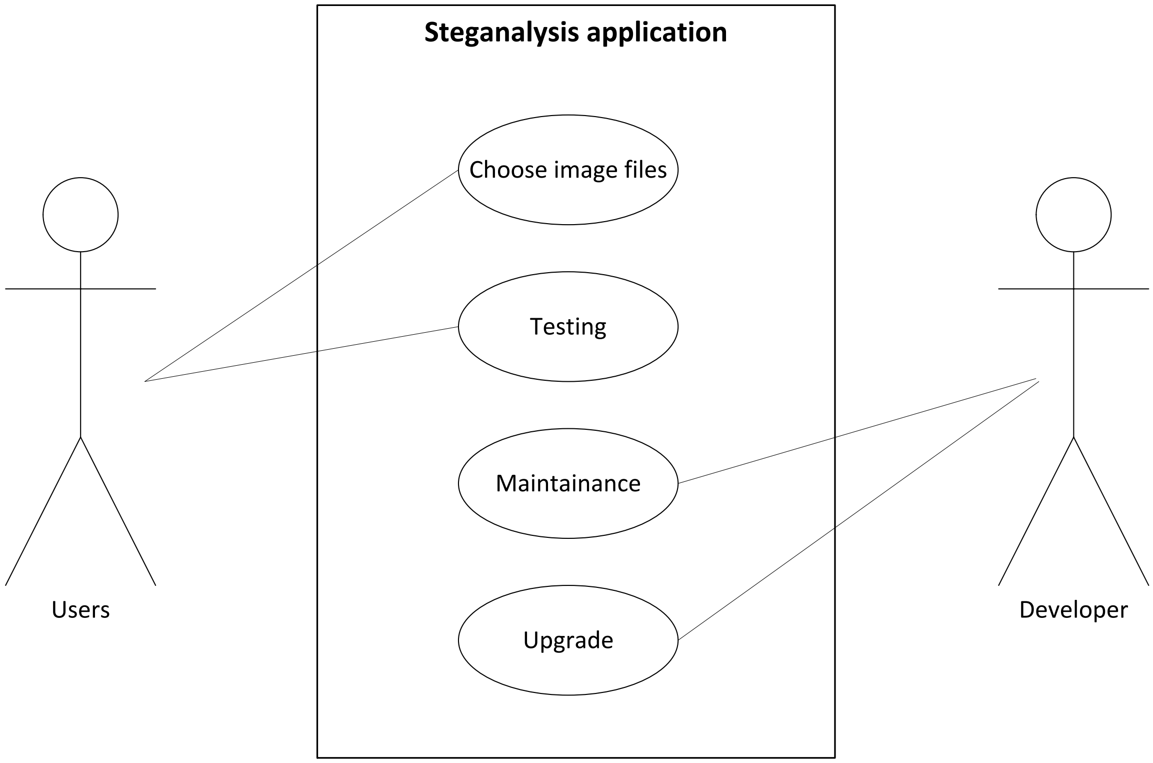


Figure3.6 Use-case diagram

Users can choose an image file to test and analyze the chosen image file to determine it is a stego-image or not. Developer can perform update and maintenance on the software.

3.3.3.4 Context Diagram

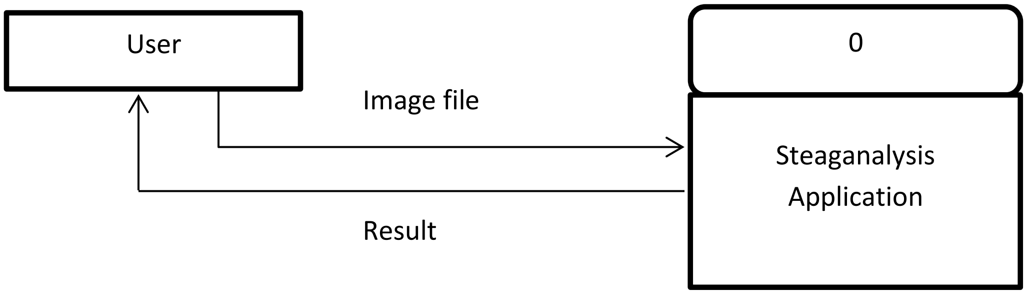


Figure3.7 Context diagram

The external entity is user. The entity is steganalysis application. User will send the image file to steganalysis application and the application will generate result to the user.

3.3.4 Implementation Phase

The system design needs to be implemented to make it a workable system by implement coding of design into computer language (programming languages). MATLAB code is used to make the functions in the application working. The program specifications are converted into computer instructions, which also known as programs. It is an important stage because the defined procedures are transformed into control specifications by the help of a computer language. A well written code will reduces the testing and maintenance time and effort. Programming tools like compilers, interpreters and MATLAB code are used for coding process. Debugging process is needed if necessary. Besides that, some tutorials and videos from YouTube are referred due to the first time use of MATLAB on develop application. Formula of PSNR and MSE are implemented for detection of LSB modification noise.

3.3.5 Verification Phase

The application must be tested to make sure it works properly and evaluate its actual functionality. The objective is to ensure that the application can functions as expected and that end user’s requirements are satisfied and fulfilled. Unit, system, performance and user acceptance tests are performed during this phase. For unit test, each unit of the source codes, operating procedures and interfaces are test independently. For user acceptance test, end users will determine whether the developed system meets the intended requirements. For performance test, responsiveness, scalability, reliability, robustness, durability, resources usage and stability of the system is tested under a particular workload. For system testing, the whole system is tested to ensure that all modules work together correctly and all functions perform well as an application or system. If there are any problems, debugging the program is needed and the bugs will be fixed and will be tested again for the bug fix. Stegoimages are created using steganography software for example JP Hide and Seek for testing purposes to test the functions and performances. When all problems are fixed or there are no problems anymore after testing by the end users, then the software can be used.

3.3.6 Maintenance Phase

During the maintenance phase, 3 things happen to the software. They are bug fixing, upgrade and enhancement. Because of some untested scenarios, the software may give errors or logical problems and these bugs have to fixed and repaired. For example, operating system or environmental changes or hardware changes may cause some changes in the programs also so some bugs and errors will appear. The system may upgrade to make it compatible with the upgraded operating system and hardware. There are always some rooms to add new features to the application so enhancement is needed to make the application become better. Feedback from the users is collected to determine the necessity of bug fixing, upgrade and enhancement.

3.4 Hardware and Software Requirements

It is very important to identify the kinds of hardware and software technology used to make sure the system can function well and maximize the performances of the system. Hardware performances like responds time and disk spaces may affect the performance of the system and the software used. Good software is needed to create a good system.

3.4.1 Hardware requirements

Hardware	Description
Compaq CQ42 (Laptop)	• Intel® Core™ i5 CPU M430 @ 2.27GHz • 2GB RAM • Window 7 professional 32 bits

Table 3.1 Hardware Description

Compaq laptop is used to create the system with the aids of the chosen software. Window 7 is also compulsory because all of the software used is compatible with it. Laptop is also used for the testing purposed of the new system. All the documentation is done by using laptop and also save them inside the laptop.

3.4.2 Software requirements

Software	Description
MATLAB	• Version R2010a • Language: MATLAB code • Compiler
Pixillion image converter	• OS: window • Version 2.54
JP Hide and Seek	• OS: window • JPEG supported
OpenStego	• OS: Window • BMP and PNG supported
Image Steganography	• OS: Window • JPG and PNG supported

Table 3.2 Software description

MATLAB is used to create the application software. Pixillion Image Converter is used to convert the image format. Image Steganography, OpenStego, and JP Hide and Seek are used to create stego image for testing purposes.

3.5 Conclusion

Peak Signal-to-Noise Ratio (PSNR) steganalysis is chose as the method used in my program because it is use for detect various LSB modification techniques. LSB is the method that most commonly used in steganography to hide message. So, it is necessary to develop a program to detect it. Agile model is chosen to aids the application development progress. For future, coding process and testing will be carried out. New requirements will be taken as guidance to improve the application.

Chapter 4:

Design and Implementation

During the design process, a GUI application is designed for detection of presence of marks in stegocontent. During implementation process, the coding process is carried out in order to produce a functional application. The application is designed using MATLAB and the languages used for coding is MATLAB code. MATLAB allows developer to create and design the graphical interface for the application. MATLAB also allows developer to compile the coding using its complier. Besides, developer can perform testing, debugging, coding processes to produce a good application. The graphical user interface of the application is user friendly and it is easy to let users understand the ways to operate each of its functions. The algorithm coding of detection of presence of marks in stegocontent is implemented into the application on the test button. The application is tested to ensure the coding and application itself is error free.

4.1 Overall Interface Layout

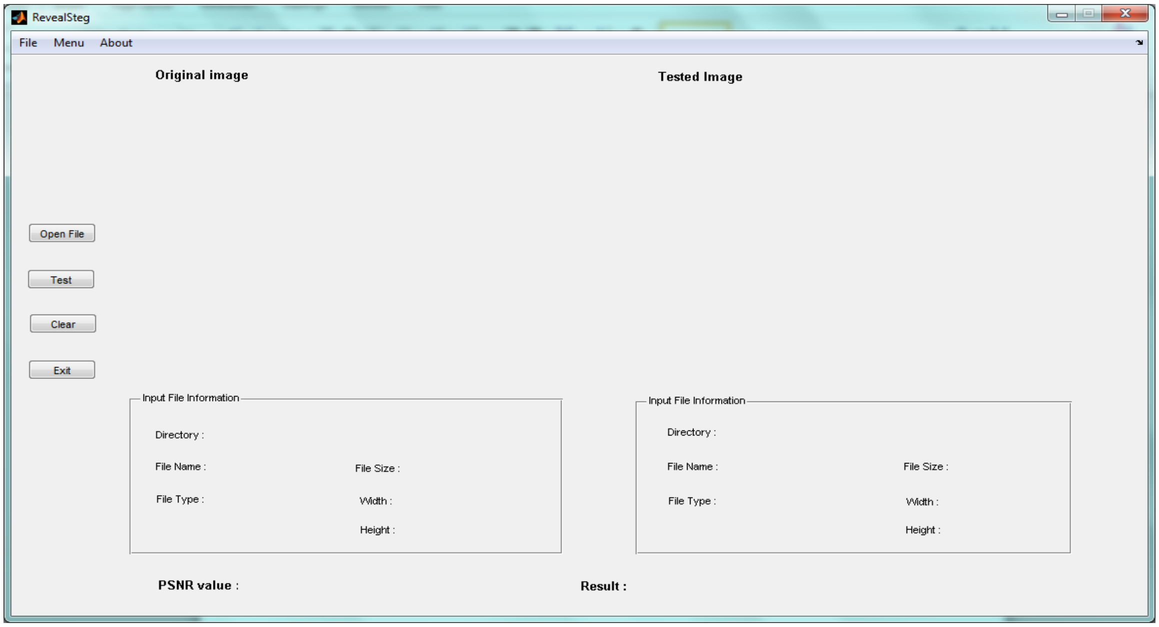


Figure4.1Overall Interface Layout.

This is the overall layout of the application. There are buttons, labels, panels, and menu bar in this interface.

4.1.1 Open File Function

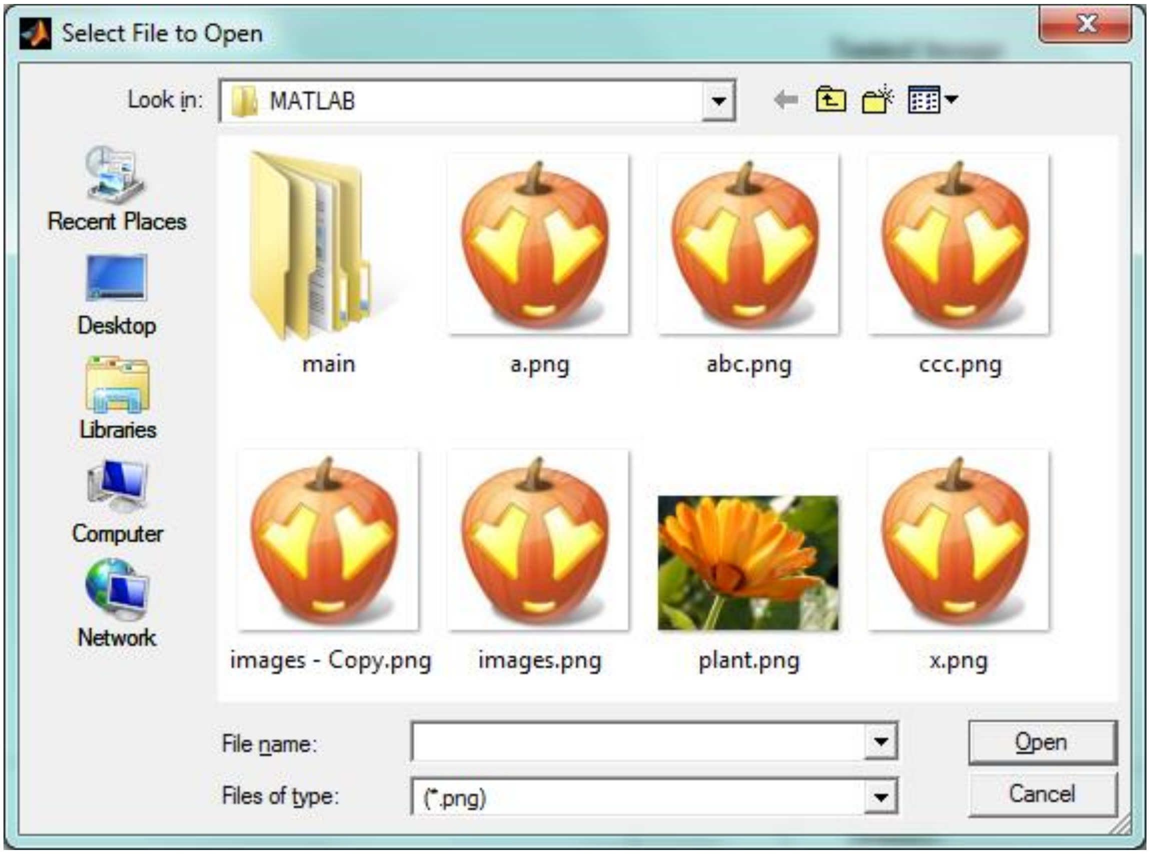
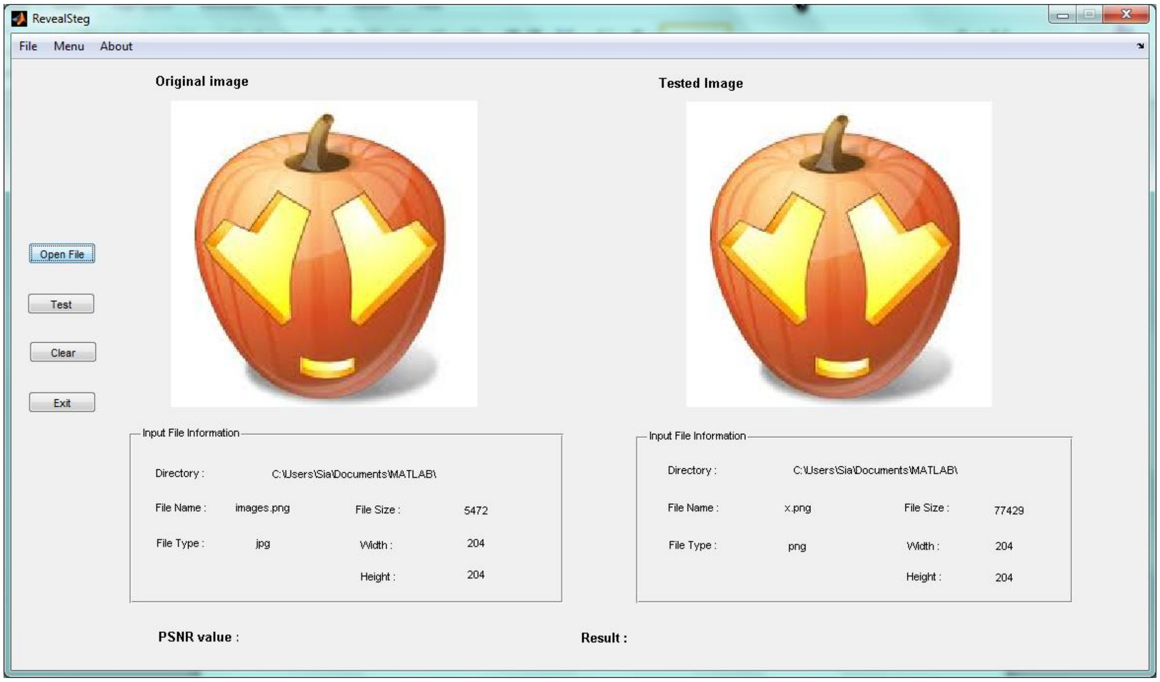


Figure 4.2 Open file layout.

```
[filename pathname] = uigetfile('*');
[filename1 pathname1] = uigetfile('*');
```

When Open File button is clicked, a file selector will pop out to let users choose images to be tested. User is required to choose image twice because one is for original image and another one is for tested image. After the images are chosen, the layout will be same as Figure 4.3.



Input File Information

Directory : C:\Users\Sia\Documents\MATLAB\

File Name : images.png File Size : 5472

File Type : jpg Width : 204

Height : 204

Input File Information

Directory : C:\Users\Sia\Documents\MATLAB\

File Name : x.png File Size : 77429

File Type : png Width : 204

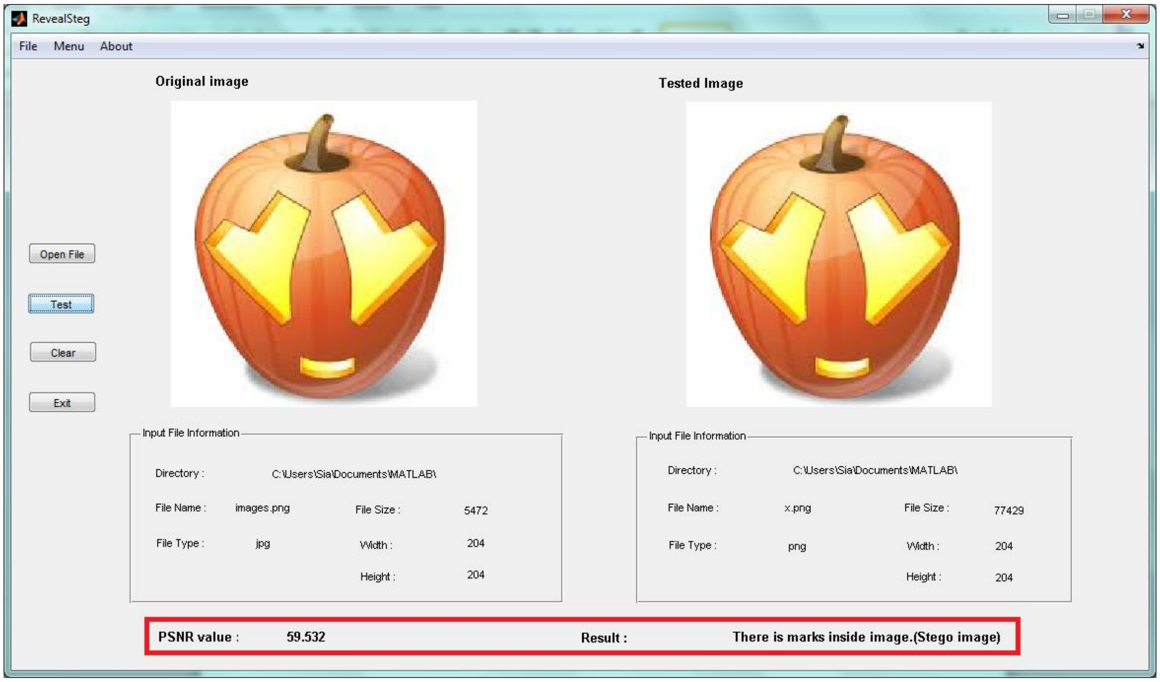
Height : 204

Figure 4.3 Interface’s layout when Open File button is clicked.

```
imshow(fullfile(pathname,filename));
axes(handles.axes1);
strcat(pathname,filename);
set(handles.DirectoryTxt,'string',pathname);
set(handles.Filenametxt,'string',filename);
image=imread(fullfile(pathname,filename));
handles.image=image;
assignin('base','filename',filename);
guidata(hObject,handles);
fileInfo = dir(fullfile(pathname,filename));
fileSize = fileInfo.bytes;
set(handles.sizetxt,'string',fileSize);
info = imfinfo(fullfile(pathname,filename));
imginfo = info.Format;
imgwidth = info.Width;
imgheight = info.Height;
set(handles.widthtxt,'string',imgwidth);
set(handles.heighttxt,'string',imgheight);
set(handles.typetxt,'string',imginfo);
```

The images chosen and their information for example directory, file name, file type, file size, width, and height is shown.

4.1.2 Test Function



Directory : C:\Users\Sia\Documents\MATLAB\
File Name : images.png File Size : 5472
File Type : jpg Width : 204
Height : 204

Directory : C:\Users\Sia\Documents\MATLAB\
File Name : x.png File Size : 77429
File Type : png Width : 204
Height : 204

PSNR value : 59.532 Result : There is marks inside image.(Stego image)

Figure 4.4 Interface’s layout when Test button is clicked.

```
filename=evalin('base','filename');
filename1=evalin('base','filename1');
original=imread(num2str(filename));
stego=imread(num2str(filename1));

if MSE == 0
     set(handles.resulttxt,'string','There is no marks detected.(Original
image)');
 else
     set(handles.resulttxt,'string','There is marks inside image.(Stego
image)');
 end
```

The PSNR (Peak Signal-to-Noise Ratio) value and result are shown.

4.1.3 Clear Function

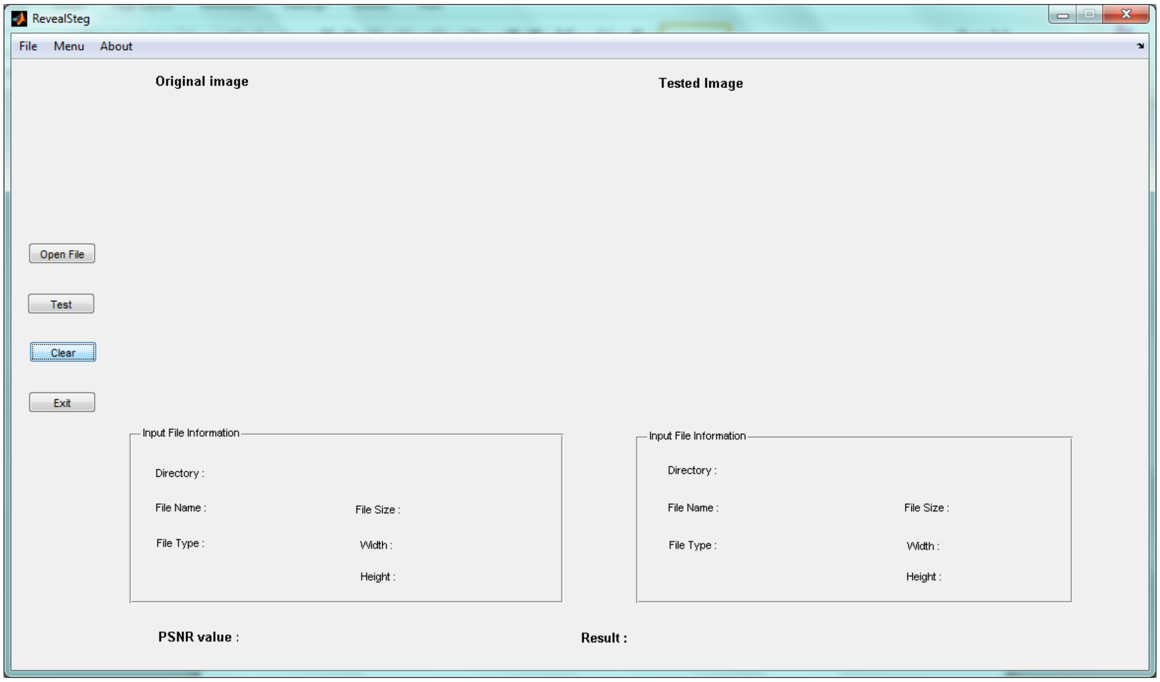


Figure 4.5 Interface's layout when Clear button is clicked.

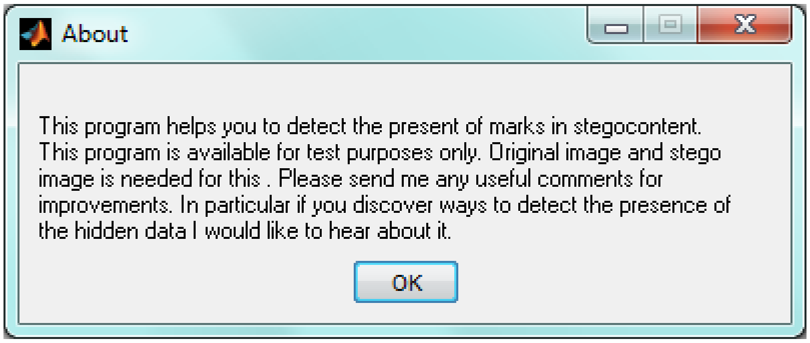
```
h=findobj(handles.axes1,'type','image');
delete(h);
i=findobj(handles.axes2,'type','image');
delete(i);
set(handles.DirectoryTxt,'string','');
set(handles.Filenametxt,'string','');
set(handles.sizetxt,'string','');
set(handles.widthtxt,'string','');
set(handles.heighttxt,'string','');
set(handles.typetxt,'string','');

set(handles.directorytxt1,'string','');
set(handles.filenametxt1,'string','');
set(handles.filesize1,'string','');
set(handles.width1,'string','');
set(handles.height1,'string','');
set(handles.filetypetxt1,'string','');

set(handles.psnrtxt,'string','');
set(handles.resulttxt,'string','');
```

All the values, strings and images are cleared.

4.1.4 About Menu Layout



This program helps you to detect the present of marks in stegocontent. This program is available for test purposes only. Original image and stego image is needed for this . Please send me any useful comments for improvements. In particular if you discover ways to detect the presence of the hidden data I would like to hear about it.

Figure 4.6 About menu layout

```
message={'This program helps you to detect the present of marks in
stegocontent. This program is available for test purposes only. Original
image and stego image is needed for this . Please send me any useful
comments for improvements. In particular if you discover ways to detect
the presence of the hidden data I would like to hear about it.'};
msgbox(message,'About');
```

The message will be shown in this layout. There is a OK button to allow users exit this layout after reading the message.

4.1.5 Exit Function

```
close all;
```

The application will be closed after the Exit button is clicked.

4.2 Algorithm Detecting Marks in Stegocontent

```
[row,col] = size(original);
size_host = row*col;
 o_double = double(original);
 s_double = double(stego);
 s = 0;
 for j = 1:size_host;
 s = s+(s_double(j) - o_double(j))^2 ;
 end
 MSE = s/size_host;
 psnr = 10*log10((255)^2/MSE);
```

This is the main algorithm of the test image function in the application. The size of the original image and tested image are recorded and convert them into double. The Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) are calculated by using the formula below:

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

Two m ×n monochrome images I and K where one of the images is considered a noisy approximation of the other where i and j is the size of the image (row and column).

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \end{aligned}$$

MAX is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, it is 255. For color images with three RGB values per pixel, the definition of PSNR is the same except the MSE is the sum over all squared value differences divided by image size and by three.

Typical values for the PSNR in edited image is between 30 and 50 dB, where higher is the values lower the noise found in image. When the two images are identical, the MSE will be zero so the value the PSNR becomes undefined or infinity.

Chapter 5:

Result and Discussion

5.1 Result Analysis

The developed application, it has met all the objectives of this project, which are:

iv. To study current steganography tools and the methods used for detection of presence marks in stegocontent.

v. To develop steganalysis tools which implement with *graphical user interface (*GUI).

vi. To verify and detect the marks in stegocontent in image files to assist investigation process.

Steganography is a technique of hiding message by embedding message within other types of media and the most popular media used is image file. However, most of the people are not familiar with steganography or they never know the existence of this technique. Currently there are actually quite many steganography tools are developed and can be found from internet easily. There are some famous steganography tools like Outguess, JP hide and seek, and OpenStego. Those steganography tools used different methods to hiding message to the target media for example replacing least significant bits (LSB), transformation domain technique (DCT), and distortion technique. As a result, there are also lots of methods used for detection of presence of marks in stegocontent. For example RS (Regular and Singular groups) steganalysis , PoV(*Pairs of Value*)-based Chi-square test, Palette checking, RQP (*Raw Quick Pairs)* method, Check JPEG compatibility, Histogram analysis, Universal blind detection, and Peak Signal-to-Noise Ratio (PSNR) analysis. Those methods perform with different algorithm and technique to detect various type of stegocontent. However, the numbers of steganalytic tools are less when compare to the numbers of steganography tools.

Although there are some steganalytic tools to detect the stegocontent, they are mostly not implemented with Graphical User Interface (GUI). Some of them need to enter command to perform their functions so many new users does not understand how to operate them and it is complex to use. As a result, an application with GUI needs to be developed to detect the presence marks in stegocontent. This application is developed using MATLAB because MATLAB can perform many types of image processing functions and also the calculation. Besides, this application is developed for all computer users.

Before create the application, design of GUI and selection of methods to detect the presence marks in stegocontent is important. The design of the GUI must user friendly and suitable for non-experience users to use. The method chose to detect the presence marks in stegocontent is Peak Signal-to-Noise Ratio (PSNR) analysis. By using this method, the modification on the LSB of the image file can be detected. This type of method is chosen because LSB modification is the most common method used in steganography. The PSNR value can detect and verify the presence of the marks in stegocontent. If the PSNR value is infinity or undefined, there are no marks in the image. However if the PSNR value is shown, that means the image had been modified and there is the marks inside. The higher the PSNR value, the lower the numbers of marks in the stegocontent. The tables below show the results of testing.

	Files Embedded					
	Text Only		Images		Document Files	
	50bytes	100bytes	206kb	613kb	30kb	200kb
PSNR values	88.54	85.96	51.45	46.60	60.09	51.79

Table 5.1 PSNR values with different types of files embedded and size

The more bytes of files or text embedded in an image, the PSNR value is less. Therefore, the noise or marks inside the stegoimge is more.

	Files Embedded		
	Text Only	Images	Document Files
Open Stego	N/A	✓	✓
Image Steganography	✓	✓	✓
JP Hide & Seek	N/A	✓	✓

Table 5.2 Detection availability according steganography software and types of files embedded

5.2 Constraints

Constraints for this project are categorized into two parts:

I. Development constraints

II. System constraints

5.2.1 Development Constrains

There are some constrains have been faced during development of the application. MATLAB codes have to learn from beginning because never tried MATLAB before in order to implement the coding to the GUI and the functions. The coding to develop the application is revised from the internet sources and all of them are open sources. It is difficult to study and understand the algorithms and formulas so this leads to more time consumed. It is difficult to implement several types of methods to detect the presence marks in stegocontent because some of them are not open sources.

5.2.2 System Constraints

This application is still a prototype and it must require MATLAB to operate it. Besides that, it must require original image to test the suspected image. This application only support for some image types. This application can only detect the LSB modification in the images. Other steganography media for example audio files and text files cannot be tested using this application.

5.3 Suggestion and Enhancement of Project Developed

There are a few enhancements that can be carried out for future improvement of the application.

I. Provides more functions to detect other types of steganography techniques.

II. Provides the detection for other types of stego media.

III. Use multiple methods to detect specific type of steganocontent to improves the accuracy of the results.

Chapter 6:

Conclusion

In conclusion, the research on detection of presence marks in stegocontent goes successfully and smoothly. Along the research, learning and understanding the steganogrphic techniques is quite challenging. Throught the study on the existing steganography tools and the steganalytic methods, the objectives and purposes of steganography is identified. Besides, many people are found that have less knowledge on steganography or not even know existence of the technique. However, there are many steganography softwares on internet that free for download but there is less applications that provide detection of marks in stegocontent. Therefore, a prototype application is designed and created which provide a GUI environment for users to detect the presence marks in stegocontent. The technique used in the prototype application is not the best technique but it can still function well to detect the marks in stegocontent. It is ready and stable enough for user to use it. The development of prototype application is completed although there are lots of constraints during the development of the application and system itself. All the activities carried out along the project is followed excatly by Gantt chart. In future, the user interface of the application will be updated to look much more better and user friendly. In addition, the application will also provides more functions to detect other types of steganography techniques and detection for other types of media. Multiple methods will be implemented into the application to detect specific type of stegocontent to improve the accuracy of the results.

References

Bennett, K. (2004). Linguistic Steganography: Survey, Analysis, and Robustness Concerns for Hiding Information in Text. Purdue University, *CERIAS Tech. Report*. pp. 2-29.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. United Kingdom: Springer Verlab.

Bhattacharyya, S. & Sanyal, G. (2012). A robust image steganography using DWT difference modulation (DWTDM). *I. J. Computer Network and Information Security*. 7(1): 27-40.

Callinan, J. and Kemick, D. (2006). Detecting Steganographic Content in Images Found on the Internet. http://www.chromesplash.com/jcallinan.com/publications/steg.pdf (19 October 2012)

Calpe, A. (2006). Steganography in Images. Retrieved from http://www.cs.ucf.edu/courses/cot4810/fall04/presentations/Steganography_in_Images.ppt/ (19 October 2012)

Chan Y. K. and Chang C. C. (2001). Concealing a secret image using the breadth first traversal linear quad tree structure. *IEEE Proceedings of Third International Symposium on Cooperative Database Systems for Advanced Applications*. pp.194-199.

Chandramouli, R. and Memon, N.D. (2006). Steganography Capacity: A Steganalysis Perspective. http://citeseer.ist.psu.edu/cache/papers/cs/27278/http:zSzzSzwww.ece.stevens-tech.eduzSz~moulizSzstegcap03.pdf/chandramouli03steganography.pdf/ (19 October 2012)

Cole, E. (2003). *Hiding in Plain Sight: Steganography and the Art of Covert Communication*. New York: Wiley, John & Sons, Incorporated.

Curran, K., & Devitt, J. M. (2008). Image Analysis for Online Dynamic Steganography Detection. *Computer and Information Science*, 1(3). 32-41.

Druid. (2006). Steganography Primer: Introduction to Steganography. Retrieved from http://druid.caughq.org/presentations/Steganography-Primer/ (19 October 2012)

Johnson, N. F., & Jajodia, S. (1998). Steganalysis: the investigation of hidden information. *Proceedings of the 1998 IEEE Information Technology Conference.* pp. 113-116.

Johnson, N.F. & Jajodia, S. (1998). Exploring Steganography: Seeing the Unseen. *Computer Journal*. pp. 26-34.

Johnson, N.F. (2000). *Information Hiding: Techniques for Steganography and Digital Watermarking*. Boston, USA: Artech House.

Krenn, R. (2000). Steganography and Steganalysis. http://www.krenn.nl/univ/cry/steg/article.pdf (19October2012)

Kumar, A., & Pooja, K. (2010). Steganography- A data hiding technique. *International Journal of Computer Applications,* 9(7), 19-23.

Lee, Y. K. and Chen L. H. (2000). High Capacity Image Steganographic Model. *IEE Proceedings Vision, Image and Signal Processing*. pp. 288-294.

Manchanda, S., Dave, M. and Singh, S. B.(2007). Customized and secure image steganography through random numbers logic. *Signal Processing: An International Journal*. 1(1): 1-16.

Mercuri, R.T. (2004). The many colors of multimedia security. *Communications of the ACM*. 47(12): 25–29.

Nagham Hamid, Abid Yahya, R. Badlishah Ahmad, & Osamah M. Al-Qershi. (2012). Image steganography techniques: an overview. *International Journal of Computer Science and Security (IJCSS).* 6(3) : 168-187.

Provos, N., Honeyman, P. (2006). Detecting Steganographic Content on the Internet. http://niels.xtdnet.nl/papers/detecting.pdf (19 October 2012)

Queirolo, F. (2006). Steganography in Images. http://www.cse.buffalo.edu/~peter/cse741/Presentations/Refs/Queirolo.pdf (19 October 2012)

Si, B. (2004). Introduction to Steganography. http://www.infosyssec.com/infosyssec/Steganography/menu.htm (19 October 2012)

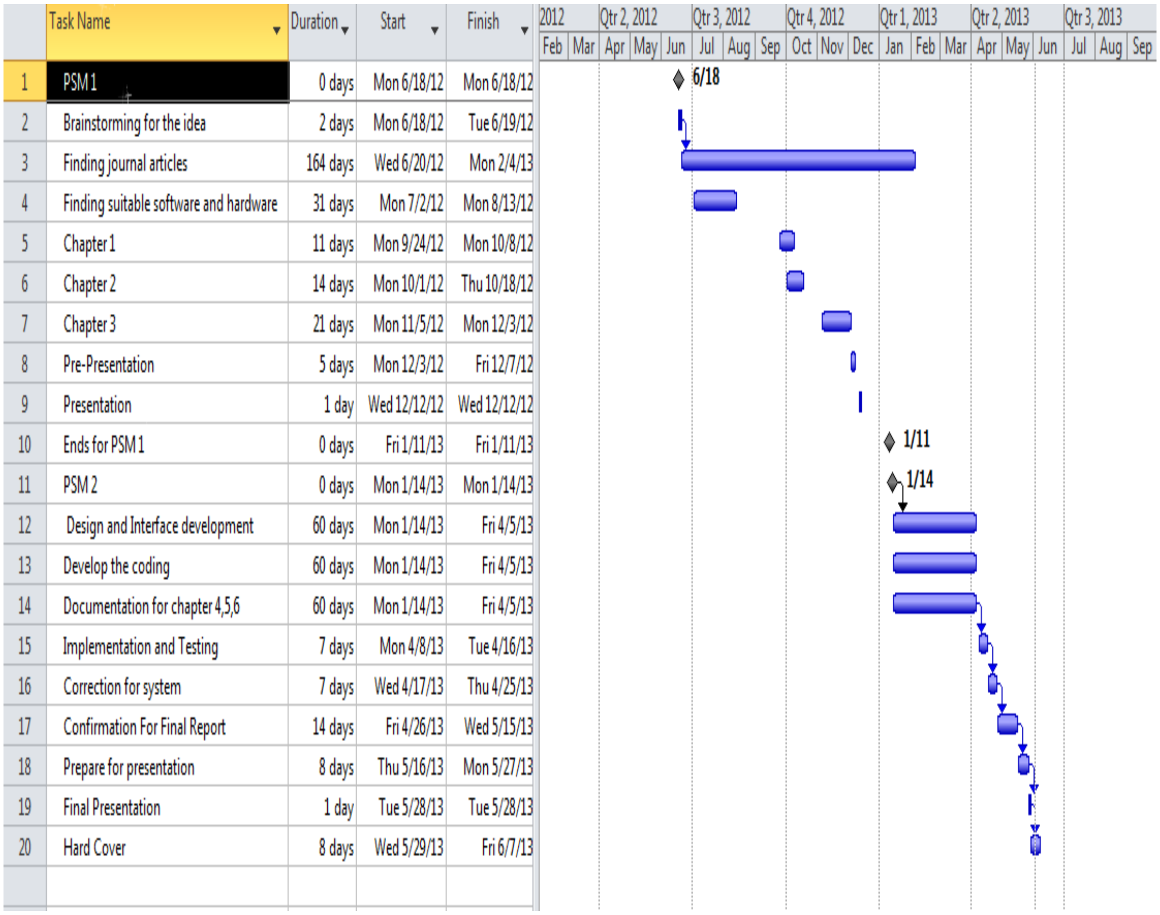
Wang, H. Q., & Wang, S. Z., (2004). Cyber warfare: steganography vs. steganalysis. *Communication of the ACM*. 47(10): 76-82.

Wayner, P. (2002). *Disappearing cryptography* (2nd ed.). San Francisco, USA: Morgan Kaufmann Publishers.

Westfeld, A., and Pfitzmann, A. (1999). *Proceedings of the Third Intl.Workshop on Information Hiding*. London, UK: Springer Verlag.

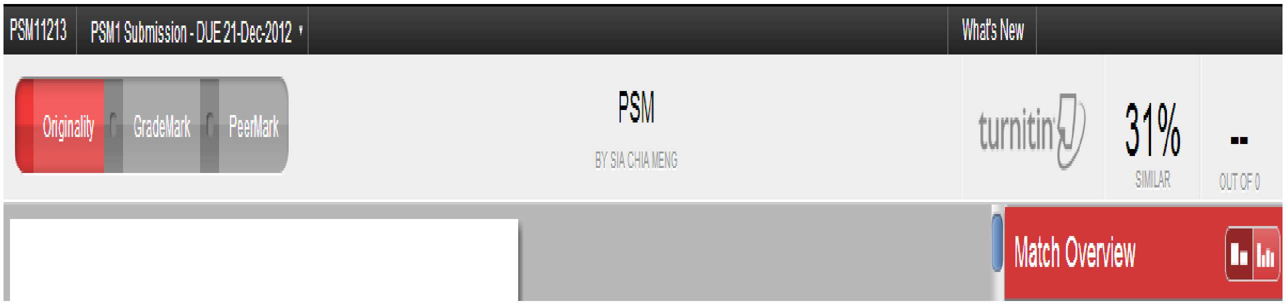
Zhang, T., Ping, X. (2006). A Fast and Effective Steganalytic Technique Against JSteg Like Algorithms. http://citeseer.ist.psu.edu/cache/papers/cs/26891/http:zSzzSzwww.rbfn.comzSzTaoZhangzSzPaperzSzacm_sac_2003_zhang.pdf/zhang03fast.pdf (19 October 2012)

Appendix A



	Task Name	Duration	Start	Finish
1	PSM 1	0 days	Mon 6/18/12	Mon 6/18/12
2	Brainstorming for the idea	2 days	Mon 6/18/12	Tue 6/19/12
3	Finding journal articles	164 days	Wed 6/20/12	Mon 2/4/13
4	Finding suitable software and hardware	31 days	Mon 7/2/12	Mon 8/13/12
5	Chapter 1	11 days	Mon 9/24/12	Mon 10/8/12
6	Chapter 2	14 days	Mon 10/1/12	Thu 10/18/12
7	Chapter 3	21 days	Mon 11/5/12	Mon 12/3/12
8	Pre-Presentation	5 days	Mon 12/3/12	Fri 12/7/12
9	Presentation	1 day	Wed 12/12/12	Wed 12/12/12
10	Ends for PSM 1	0 days	Fri 1/11/13	Fri 1/11/13
11	PSM 2	0 days	Mon 1/14/13	Mon 1/14/13
12	Design and Interface development	60 days	Mon 1/14/13	Fri 4/5/13
13	Develop the coding	60 days	Mon 1/14/13	Fri 4/5/13
14	Documentation for chapter 4,5,6	60 days	Mon 1/14/13	Fri 4/5/13
15	Implementation and Testing	7 days	Mon 4/8/13	Tue 4/16/13
16	Correction for system	7 days	Wed 4/17/13	Thu 4/25/13
17	Confirmation For Final Report	14 days	Fri 4/26/13	Wed 5/15/13
18	Prepare for presentation	8 days	Thu 5/16/13	Mon 5/27/13
19	Final Presentation	1 day	Tue 5/28/13	Tue 5/28/13
20	Hard Cover	8 days	Wed 5/29/13	Fri 6/7/13

Appendix B


PSM11213
PSM1 Submission - DUE 21-Dec-2012
What's New
Originality
GradeMark
PeerMark
PSM
BY SIA CHIA MENG
turnitin
31%
SIMILAR
--
OUT OF 0
Match Overview