IMAGE STEGANOGRAPHY

**APPLICATION DEVELOPMENT – II**

**SOFTWARE DESIGN SPECIFICATION**

*Submitted in partial fulfilment for the award of the degree*

*of*

**Master of Technology**

***in***

**Information Technology**

*by*

**Bhuvana Chandra Pathi**

**(17MIN0618)**





**School of Information Technology and Engineering**

September 2018



**School of Information Technology and Engineering**

**DECLARATION BY THE CANDIDATE**

I hereby declare that the thesis entitled **“IMAGE STEGANOGRAPHY”** submitted by me to Vellore Institute of Technology, Vellore, in partial fulfillment of the requirement for the award of the degree of **Master of Technology** in **Information Technology** is a record of bona fide APD work carried out by me. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

**Place**: Hyderabad

**Date**: 25-06-2020 **Signature of the Candidate**

### table of contents

### INTRODUCTION…………………………………………………5

## ABOUT WIPRO TECHNOLOGIES……………………………………………………………….5

## PROJECT PURPOSE……………………………………………………………………………….5

## PROJECT SCOPE…………………………………………………………………………………..8

## PROJECT OVERVIEW…………………………………………………………………………….8

### SYSTEM OVERVIEW……………………………………………11

## IMPLEMENTATION……………………………………………………………………………….11

## BASE64 CONVERSION……………………………………………………………………………11

## EMBEDDING……………………………………………………………………………………….14

## EMBEDDING PROCESS…………………………………………………………………………...14

### SYSTEM ARCHITECTURE……………………………………..18

### ARCHITECTURE…………………………………………………………………………………...18

### DECOMPOSITION DISCRIPTION………………………………………………………………...22

### SYSTEM RESOURCES……………………………………………………………………………..26

### SYSTEM REQUIREMENTS……………………………………………………………………......26

### DATA DESIGN…………………………………………………….27

### DATA DISCRIPTIONS……………………………………………………………………………...27

### ER DIAGRAM……………………………………………………………………………………….27

### HUMAN INTERFACE DESIGN…………………………………29

### OVERVIEW OF USER INTERFACE………………………………………………………………29

### SCREEN IMAGES………………………………………………………………………………….30

### REQUIREMENT MATRIX………………………………………36

### TECHNOLOGY OVERVIEW……………………………………………………………………....36

### FUTURE WORKS…………………………………………………………………………………...40

### APPENDICES……………………………………………………..41

## INTRODUCTION

### ABOUT WIPRO TECHNOLOGIES

Wipro Technologies is a global provider of software services. It is the world's first SEI-CMM level 5 company as well as India's largest publicly traded IT Company. Wipro has a solid track record over the last decade with over a hundred satisfied customers across US, UK, Canada, Europe and Japan.

Wipro Technologies is,

* ISO 9001 Certified
* The world’s first SEI-CMM Level 5 Software Services Company
* PCMM Level 5 Company

Wipro Technologies has adopted Six Sigma practices across the organization. Wipro Technologies is the global technology services division of Wipro Ltd. Wipro Technologies offer a full portfolio of services across industries, delivering measurable business benefits the customers with six-sigma consistency.

Wipro is a Service Organization engaged in a wide range of Product as well as consultancy services. Customers and People management are the focus areas of Wipro.

### PROJECT PURPOSE

One of the reasons that intruders can be successful is most of the information they acquire from a system is in a form that they can read and comprehend. Intruders may reveal the information to others, modify it to misrepresent an individual or organization, or use it to launch an attack. One solution to this problem is, through the use of steganography. Steganography is a technique of hiding information in digital media. In contrast to cryptography, it is not to keep others from knowing the hidden information, but it is to keep others from thinking that the information even exists.



#### **What is Steganography?**

Steganography is the practice of hiding private or sensitive information within something that appears to be nothing out to the usual. Steganography is often confused with cryptology because the two are similar in the way that they both are used to protect important information. The difference between two is that steganography involves hiding information, so it appears that no information is hidden at all. If a person or persons views the object that the information is hidden inside of he or she will have no idea that there is any hidden information, therefore the person will not attempt to decrypt the information.

What steganography essentially does is exploit human perception, human senses are not trained to look for files that have information inside of them, although this software is available that can do what is called Steganography. The most common use of steganography is to hide a file inside another file.

#### **History of Steganography:**

Throughout history Steganography has been used to secretly communicate information between people. Some examples of use of Steganography is past times are:

1. During World War 2 invisible ink was used to write information on pieces of paper so that the paper appeared to the average person as just being blank pieces of paper. Liquids such as milk, vinegar and fruit juices were used, because when each one of these substances are heated they darken and become visible to the human eye.
2. In Ancient Greece they used to select messengers and shave their head, they would then write a message on their head. Once the message had been written the hair could grow back. After the hair grew back the messenger was sent to deliver the message, the recipient would shave off the messenger’s hair to see the secrete message.
3. Another method used in Greece was where someone would peel wax off a tablet that was

#### **Why This Steganography?**

This technique is chosen, because this system includes not only imperceptibility but also un-delectability by any steganolysis tool.

Steganography become more important as more people join the cyberspace revolution. Steganography is the art of concealing information in ways that prevents the detection of hidden messages. Steganography include an array of secret communication methods that hide the message from being seen or discovered.

Due to advances in ICT, most of information is kept electronically. Consequently, the security of information has become a fundamental issue. Besides cryptography, steganography can be employed to secure information. In cryptography, the message or encrypted message is embedded in a digital host before passing it through the network, thus the existence of the message is unknown. Besides hiding data for confidentiality, this approach of information hiding can be extended to copyright protection for digital media: audio, video and images.

The growing possibilities of modern communications need the special means of security especially on computer network. The network security is becoming more important as the number of data being exchanged on the internet increases. Therefore, the confidentiality and data integrity are requires to protect against unauthorized access and use. This has resulted in an explosive growth of the field of information hiding

Information hiding is an emerging research area, which encompasses applications such as copyright protection for digital media, watermarking, fingerprinting, and steganography.

In watermarking applications, the message contains information such as owner identification and a digital time stamp, which usually applied for copyright protection.

Fingerprint, the owner of the data set embeds a serial number that uniquely identifies the user of the data set. This adds to copyright information to makes it possible to trace any unauthorized use of the data set back to the user.

Steganography hide the secrete message within the host data set and presence imperceptible and is to be reliably communicated to a receiver. The host data set is purposely corrupted, but in a covert way, designed to be invisible to an information analysis.

### PROJECT SCOPE

The primary idea behind developing this project is to protect confidential data from an intruder’s counter-attacks and to block the intruder through various levels in his/her attacks. A new tool has been developed with a combination of cryptographic encryption and steganography encryption for its implementation. The developed steganography tool has a sender’s segment that can take a message, a password and a cover image as input and give a stego-image as output that has message embedded in it. On the other hand, it also has a receiver’s segment where the receiver inputs the stego-image and the same password is used by the sender as input to get the sender’s message as output. This project is developed for hiding information in any image file. The scope of the project is implementation of steganography tools for hiding information includes any type of information file and image files and the path where the user wants to save Image and extruded file.

### PRODECT OVERVIEW

The word steganography comes from the Greek “Seganos”, which mean covered or secret and – “graphy” mean writing or drawing. Therefore, steganography means, literally, covered writing. It is the art and science of hiding information such its presence cannot be detected, and a communication is happening. A secret information is encoding in a manner such that the very existence of the information is concealed. Paired with existing communication methods, steganography can be used to carry out hidden exchanges.

The main goal of this projects it to communicate securely in a completely undetectable manner and to avoid drawing suspicion to the transmission of a hider data. There has been a rapid growth of interest in steganography for two reasons:

The publishing and broadcasting industries have become interested in techniques for hiding encrypted copyright marks and serial numbers in digital films, audio recordings, books and multimedia products

Moves by various governments to restrict the availability of encryption services have motivated people to study methods by which private messages can be embedded in seemingly innocuous cover messages.

The basic model of steganography consists of Carrier, Message and password. Carrier is also known as cover-object, which the message is embedded and serves to hide the presence of the message.

Basically, the model for steganography is shown on following figure:

Cover-object, C

Message, M

Stego-key, K

**F(X,M,K)**

Stego Object, Z

Message is the data that the sender wishes to remain it confidential. It can be plain text, cipher text, other image, or anything that can be embedded in a bit stream such as a copyright mark, a covert communication, or a serial number. Password is known as stego-key, which ensures that only recipient who know the corresponding decoding key will be able to extract the message from a *cover-object*. The *cover-object* with the secretly embedded message is then called the *Stego-object*.

Recovering message from a *stego-object* requires the *cover-object* itself and a corresponding decoding key if a *stego-key* was used during the encoding process. The original image may or may not be required in most applications to extract the message.

There are several suitable carriers below to be the *cover-object:*

* Network protocols such as TCP, IP and UDP
* Audio that using digital audio formats such as wav, midi, avi, mpeg, mpi and voc
* File and Disk that can hides and append files by using the slack space
* Text such as null characters, just alike morse code including html and java
* Images file such as bmp, gif and jpg, where they can be both colour and grey-scale.

In general, the information hiding process extracts redundant bits from *cover-object.* The process consists of two steps:

* Identification of redundant bits in a *cover-object.* Redundant bits are those bits that can be modified without corrupting the quality or destroying the integrity of the *cover-object.*

Embedding process then selects the subset of the redundant bits to be replaced with data from a secret message. The *stego-object* is created by replacing the selected redundant bits with message bits.

## SYSTEM OVERVIEW

## IMPLEMENTATION

## The disadvantage with existing steganographic systems is their simple data embedding procedure. Because of this single level encryption of information the counter attack is being done at one level only and by applying all possible counter techniques blindly it may result in compromising the data. Our system is intended to encrypt data through multiple levels.

## BASE64 CONVERSION

## As part of multi-level approach the base64 conversion is taken as the first step. The intended purpose of base64 mechanism is to convert any unreadable data into a readable format. As mentioned earlier, an image to a computer is just a stream of bytes. Each byte is composed of 8-bits, and each bit is capable of storing one value, either ‘0’ or ‘1’ which we all know as ASCII standard. The ASCII standard is an 8-bit character set which has 256 (28 ) characters that form text. Among those 256 characters, only few are readable and printable, but the idea here is to generate a readable text, which could be done by decreasing the number of character set [6] [11]. To reduce the character set, the Base64 mechanism reads only 6-bits as a character instead of 8-bits. Now we will have only 64 (26 ) characters which are represented by upper case alphabet (26), lower case alphabet (26), numbers 0-9 (10) and symbols ‘+’,’/’ that is a total of 64 characters. But every byte has 8- bits, so reading just 6-bits would be troublesome. Using 3 bytes 24-bits are formed and these 24-bits are chopped into 4 equal parts of 6-bits each.

#### **Encoding:**

## To reduce the number of characters, 8-bits are reduced to 6-bits by considering 3 characters at a time. The actual translation follows mapping of values from 0 to 63 characters (A-Z, a-z, 0-9, +, /). For this, a chart is used as shown in figure.

## 

## For example, consider a string “Hello World!” .Base64 interpretation for given string is as follows:

## 1. Convert the characters to binary.

## 2. "Hel" is 01001000 01100101 01101100 in binary. (Notice that there are 24 bits).

## 3. Convert the 24 bits from three 8 bit groups to four 6 bit groups. 01001000 01100101 01101100 becomes 010010 000110 010101 101100.

## 4. Convert each of the four 6 bit groups into decimal.

## 010010 = 18

## 000110 = 6

## 010101 = 21

## 101100 = 44

## 5. Use each of the four decimals to look up the base64 character code.

## 18 = 'S'

## 6 = 'G'

## 21 = 'V'

## 44 =‘s’ 6.

## You now have your first three ASCII characters ("Hel") encoded as base64 ("SGVs").

#### **Decoding:**

#### The decoding process is relatively simple process. The most important thing here is that while encoding 8-bits were used for each character. But now for each character only 6- bits should be used, so to decode the information 4 characters are to be considered at a time.

#### For example decoding of string “YmFz” is as follows:

1. Convert the base64 characters to binary. (Remember to use 6 bit binary!) "YmFz" is 011000 100110 000101 110011 in binary.

2. Convert the 24 bits from four 6 bit groups to three 8 bit groups. 011000 100110 000101 110011 becomes 01100010 01100001 01110011.

3. Convert each of the three 8 bit groups into decimal.

01100010 = 98

01100001 = 97

01110011 = 115

4. Use each of the three decimals to look up the ASCII character for that value.

98 = 'b'

97 = 'a'

115 = 's'

We now have the first four base64 characters ("YmFz") decoded as ASCII ("bas").

In the same manner, data of any format can be encoded and decoded on the basis of 64 (26 ) characters.

## EMBEDDING

Embedding is a process that inserts the bits of information into the byte array stream of cover image. The following describes the process. Insertion of data should not change the information. That means, after inserting information into the cover image, and when the information is retrieved by receiver from image, the retrieved information should be same as the inserted information [7] [8] [10]. The information to be inserted is encrypted twice before insertion, so the double encrypted information is processed first in order to keep the information consistent. The main problem could be only with the characters followed by symbol ‘\’. Some of the characters followed by back slash ‘\’ could form an escape sequence character. To remove this affect a single backslash is replace by multiple backslashes.

The cover image is processed and interpreted in the form of bytes. That is, information of each pixel of the cover image is stored in the form of a byte for corresponding RGB values [8], [10]. The last bit of each interpreted byte of all the pixels is replaced with one bit of information that is to be inserted. In the same way, each and every bit of information that are be hidden is inserted into available pixels of the cover image till the end of information.

#### **Threshold**:

The insertion of bits of information is not straight-forward, but is done by selecting appropriate pixels to insert. That means, a threshold value is given as input by the user. And the pixel that satisfies the threshold (pixels value is greater than the threshold) is eligible for insertion. Before checking the threshold value of a pixel, its position is also considered to be even [7] in terms of row and the column index. Finally any pixel that is in an even position and whose value is greater than the user supplied threshold value is used for data to be inserted in it.

### EMBEDDING PROCESS

The following steps describe the embedding process:

1. Process the information that has to be inserted to remove escape sequences. The code written below exhibits this operation:

information = "\\" + information.replaceAll("\\\\", "\\\\\\\\") + " \\";

1. The cover image is processed in the form of bytes by the colors of pixels and stored in the form of an array.

This process is implemented with following code:

for (int i = 0, k = 0, l = 0; i < image.getWidth(); i++)

{

for (int j = 0; j < image.getHeight(); j++, k += 3, l++)

{

rgb = image.getRGB(i, j);

bytes[k] = (rgb >> 16) & 0xFF; bytes[k + 1] = (rgb >> 8) & 0xFF;

bytes[k + 2] = (rgb >> 0) & 0xFF; alpha[l] = (rgb >> 24) & 0xff;

}

}

1. Check the available size of the cover image to insert information and throw an exception if there is no enough space available. To make it work, following lines are used in code:

if ((bytes.length - 2) / 8 < information.length())

{

System.out.println("Secret Message exceed the threshold"); throw new

NotEnoughSpaceException("Information to long for Picture");

}

Now information and cover image both are in bytes form, so filtering of pixels and

insertion of information can be done.

1. To insert a value into a last bit of byte, the pixel should be in even position both in row and column. And also the value of the pixel should be greater than the input threshold. The following lines of code give a brief look at the internal operation:

for (int i = 0, k = 0; i < information.length(); i++)

{

int cur = information.charAt(i);

for (byte j = 0; j < 8; j++, k+=2) // filter by position (even position only)

{

if (bytes[k] > ThresholdBean.getThreshold()) // threshold should be satisfied bytes[k] = setLastBit(bytes[k],((cur & 1 << 7 - j) >> 7 - j) == 1);

}

}

1. This insertion is done until the end of information or till the end of last pixel of cover image. f) The process of retrieving information is done in similar manner. The last bit of a pixel satisfying both position and threshold conditions is read into a variable. The following code describes the process:

for (int i = 0; i < bytes.length;)

{

cur = 0;

for (int j = 0; j < 8 && i < bytes.length; j++, i+=2) // i value double increment

{

if (bytes[i] > ThresholdBean.getThreshold())

cur |= (bytes[i] & 1) << 7 - j;

}

1. The variable ‘cur’ is appended to a character variable to get valid information information.append((char) cur);

## SYSTEM ARCHITECTURE

### ARCHITECTURE

## Figure represents the architecture that is implemented. The modules of the steganographic tool are also included in the architecture. The user can either be the sender or the receiver.

## 

## A user is able to do all the operations as shown in figure manually or automatically by using auto-mode. Both the manual and auto-mode follows the same system flow

### 3.1.1 Sequence Diagram

## Sequence diagrams are part of the UML and are used to model the interactions between the actors and the objects within a system.

## A sequence diagram shows the sequence of interactions that take place during a particular use case or use case instance.

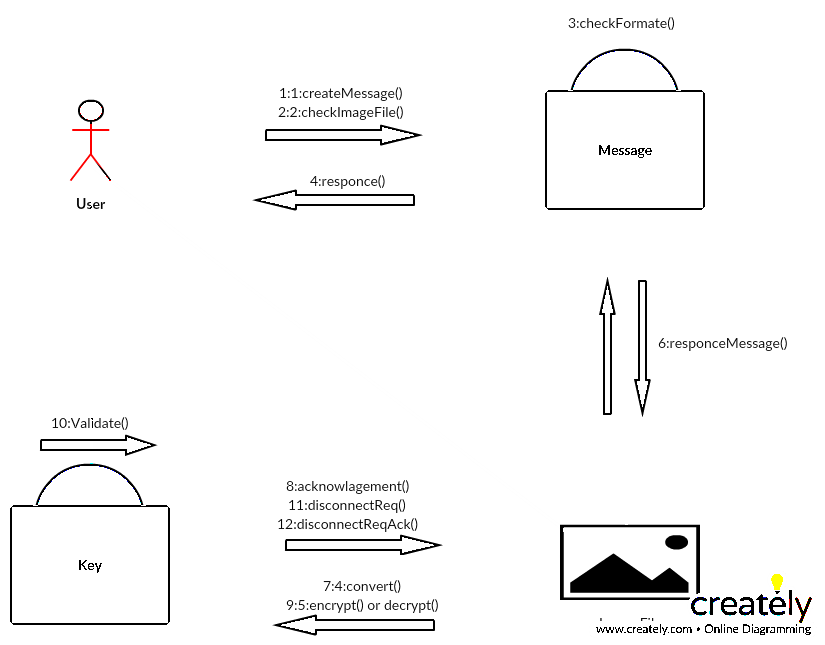
## The objects and actors involved are listed along the top of the diagram, with a dotted line drawn vertically from these.

## Interactions between objects are indicated by annotated arrows.

## 

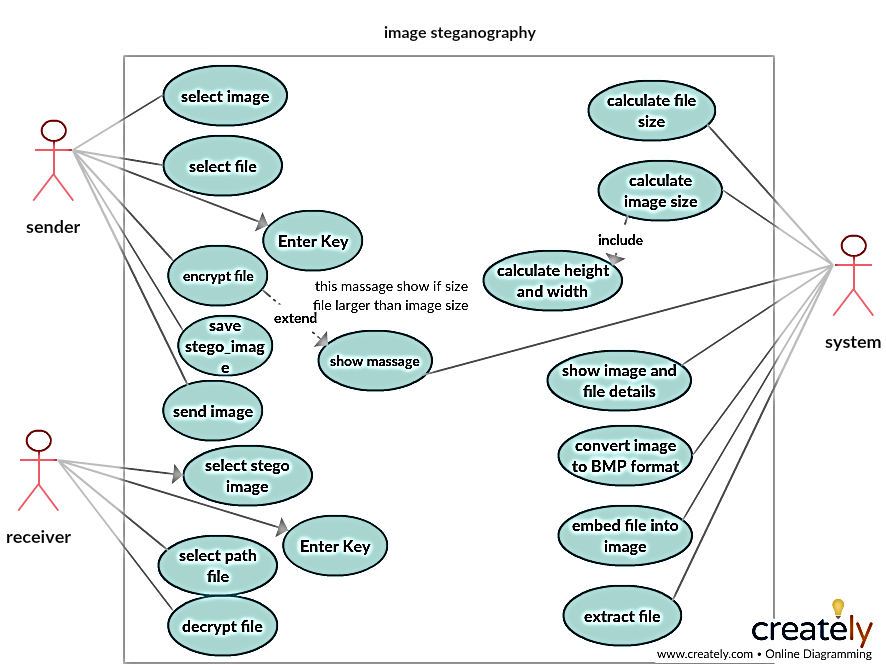
### 3.1.2 Collaboration Diagram

Displays an interaction organized around the objects and their links to one another. Numbers are used to show the sequence of messages.

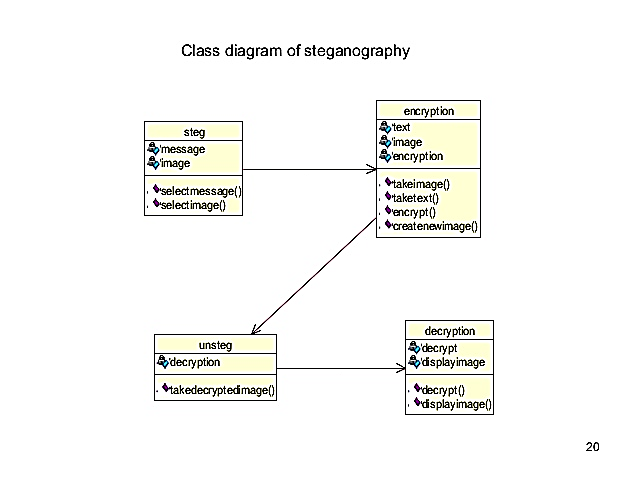


### 3.1.3 Use Cases

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.



### 3.1.4 Class /Object Diagram

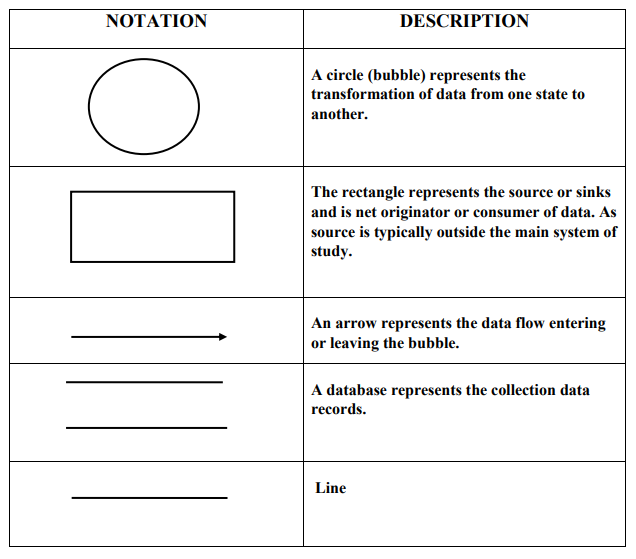


### DECOMPOSITION DISCRIPTION

### 3.2.1 Data Flow Diagrams

The data flow diagram is a way of expressing system requirements in a graphical form. A DFD also has known a bubble chart as purpose of clarifying system requirements and identifying major transformations that will become programs in design.

* DFD Notations



* Process:

A circle or a Bubble represents a process that transforms data from one form to another by performing some tasks with the data. The process name must be given a general idea of its function

* Data Flow:

1. An arrow represents data flow. It represents the path over which data travels in the system. A data flow can move between processes, flow into or out of data stores, to and from external entities. It must be given a name above the arrowhead showing the direction of flow.

2. External Entity: A rectangle, which defines the source or destination of system data also called as external entity. An external entity is not responsible for any task performed by the system.

3. Data Stores: Two horizontal parallel lines represents data store. A data store is a place where data is held temporarily from one transaction to the next or is stored permanently.

4. Data flow diagrams describe what data flows (logical) rather than how they are processed, so it does not depend on hardware, software, and data structure or file organization.

* Rules for constructing DFD:

1. Process should be named for easy understanding.

2. The direction of flow, top to bottom and from left to right should be specified. The direction flow should not allow any kind of loops.

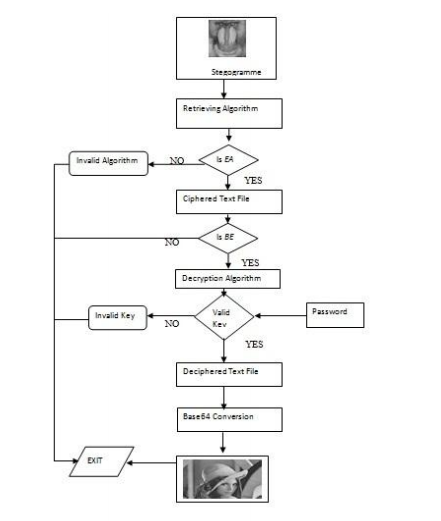
### 3.2.2 Encryption Process

## Figure shows the flow chart with the encryption part of the tool. The purpose of this project is to hide an image in other image, so a secret image would be an input. At first, the secret image is converted to a text file using Base64 conversion. Then the generated text file is encrypted with a password based encryption algorithm to generate an encrypted text file called cipher text. Using a customized embedding algorithm, cipher text is embedded on to a cover image. The output is the stegogramme (a cover image with a secret message embedded in it).

## 

**3.2.3 Decryption Process**

Figure shows a flow chart with the decryption part of the project. To read the hidden message (secret image), the stegogramme has to be decrypted. So, the stegogramme is used as input to the retrieving algorithm. If the retrieving algorithm is not the same as the embedding algorithm, there is no way that the correct output can be obtained. The correct output from retrieving algorithm is the cipher text is used as input to the decryption algorithm. This decryption algorithm is the same as the encryption algorithm; otherwise the secret message cannot be determined. And also the decryption algorithm takes a key (password) to generate plaintext.



The correct password has to be used to get plaintext. Then, that plaintext is given as input to the Base64 converter to reconstruct the secret image for the plaintext.

### 3.3. System Resources

### C# is used to implement the proposed steganographic application and as expected, the application can be used for any platform that supports Windows XP or above. For the sake of development and testing, the following hardware and software systems are used. The hardware and software involved are as follows.

### • Intel i3 core

### • Windows 10

### • C# 3.0

### • Visual Studio 2012

### 3.4. System Requirement

### Processor should contain at least 1.0 GHz speed for better performance and System required 8GB hard disk space for the Software and 512MB free RAM space for the application; however RAM space varies because of the image operations and image size.

## DATA DESIGN

### 4.1 DATA DISCRIPTION

#### Database Design maintains the data required by the System. One of the key design issues involved in the database is the distribution of data in a way that minimizes transaction traffic. Another key design issue is the choice of the database management System. Database tables used are described in the following sections. The new database design has been suggested to overcome the non-integration problems that were in the existing system. The entire system will use single integrated database. This gives the synergism to the system.

### ER DIAGRAM

The entity relationship diagram depicts the various relationships among entities, considering each objective as an entity. Entity represents the as ellipse and relationship is represented as decision diamond box. It depicts the relationship between data objects. The ERD is notation that is used to conduct the data modelling activity.

### 4.2.1 Entity:

The entity is refers to the thing which we want to store the information. An entity may be an object with physical existences (for example a person, car house etc.) or it may be an object with conceptual existences (for example a company, job etc.). Entity is an elementary basic building block of storing information about business process.

### 4.2.2 Attributes:

Attributes are the particular properties that describe the entity. Attributes are elementary pieces of information attached to an entity.

### 4.2.3. Relationship:

A relationship is a named connection or association between entities and used to relate two or more entities with some common attributes.

#### The basic notations used in ER diagram are:

#### 

## HUMAN INTERFACE DESIGN

### 5.1. OVERVIEW OF USER INTERFACE

### 5.1.1 Main Window

### It is written in C#. It is used to open the operational window or frames to perform set of operations on images using the specified techniques. It consists of basic window operations like closing and minimizing but restoring or rescaling is not possible. This main window has two tabs encode and decode which allows the user to perform his task.

### 5.1.2 Encode

### Tab *The encode* tab directs the user through three windows: Base64, Encrypt and Embed while all these windows can be executed individually. These windows were developed using C# as well as .NET framework. Use of all these three windows is not mandatory; they can be used based upon user’s security requirements. As the name of this tab says, windows involved in this tab are used in the process of encoding an input. A user can perform base64 conversion of a message, password based encryption of a message and embedding a message into a cover image. This window consists of all kinds of operations related to encoding, which are required for the developed steganographic tool.

### 5.1.3 Decode

### Tab It is purely written in C#. Similar to “encode” tab, this tab has three windows: “base64”, “decrypt” and “retrieve” which are the respective complementary decoding methods for encoding operations on “encode” tab. 25 Operations under this tab are to be used by the receiver to read/view the secret message.

### 5.1.4 Auto mode Tab

### This tab is for automatic functioning of both “encode” and “decode” tabs. That is, all the three operational windows of “encode” and “decode” tabs are integrated into a single window “encode” window and another single “decode” window respectively.

### 5.2. SCREEN IMAGES

### 

### Figure, shows the home screen or main window of the developed tool. It shows three tabs; one is encode that works on sender’s side, one is decode tab that works on receiver’s side. The third tab is auto mode tab that works for both the sender and the receiver. Using this tab all the sender’s operations can be done through one window as well as the receiver’s tab also. Communication begins from the sender’s side only when a sender hides his secret image in the cover image, and transmits the stego-image to the receiver.

### 

### Figure, shows the three operations that a sender can perform. According to the proposed model, the sender starts with base64 operation where a user is supposed to give his/her secret image as input. At the end of this operation user will have a text file as an output (image converted into text). The second option is the encryption operation, here text file is given as input and the expected output is an encrypted file. Finally the encrypted file is embedded in the cover image using the Embed option and the user (here sender) will end up with a stego-image as outcome.

### 

### Figure shows the available operations for the receiver. According to the proposed model, the receiver starts with the retrieve option. This tab shows the operation that works in a quite reverse manner to sender’s operations. The receiver uses a stego-image as his first input and extracts the embedded information using the “retrieve” operation. According to the proposed model, the retrieved information is encrypted information. A user is able to decrypt the encrypted file using decrypt operation and gets a text file as output. Finally the generated text file is converted to an image using base64 option that is a base64 decoding is done on the text file to generate the secret image as output.

### 

### Figure shows the operations available to the user. The encode option performs all the operations under the encode tab, that is, base64 encoding, encryption and finally embedding. These are done automatically by taking the required input at once to give the stego-image as output. The decode option performs all the operations under the decode tab, that is, retrieve, decrypt and finally base64 decoding. These three operations are also done automatically by taking all the inputs at a time. Finally the user (here receiver) will have the secret image as output.

### 

### Figure shows the encoding window with auto mode. Here the user has to give his secret message as input. User will be able to browse throught the computer to select his/her secret image using ‘select source’ button. To hide the secret image user requires a cover image, he/she will be able to select the cover image by browsing the computer using the ‘select socerimage’ button. The final location to save the output file should also be selected by the user using ‘select destination’ button. As the encryption here is a password based encryption, a password with a minimun of 8 characters should also be given as input. A final stego-image will be generated after giving the threshold value and clicking the ‘embed secret’ button.

### 

### Figure shows the decryption process window for the user in auto-mode, where the reverse process to the sender’s operations is done automatically. The input here should be a stego-image which can be given as input by browsing the computer. The destination location for the output file to save should also be given by the user. The password used and the threshold value should be similar to that of the password and threshold used by the sender respectively. Final image will be generated by clicking ‘retrieve secret’ button.

## REQUIREMENT MATRIX

### 6.1. TECHNOLOGY OVERVIEW

### 6.1.1 C# .Net

C# is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006). C# is one of the programming languages designed for the Common Language Infrastructure. .NET is built on the Windows Server System to take major advantage of the OS and which comes with a host of different servers which allows for building, deploying, managing and maintaining Web based solutions. The Windows Server System is designed with performance as priority and it provides scalability, reliability, and manageability for the global, Web-enabled enterprise. The Windows Server System integrated software products are built for interoperability using open Web standards such as XML and SOAP. .NET is a "Software Platform". It is a language-neutral environment for developing rich .NET experiences and building applications that can easily and securely operate within it. When developed applications are deployed, those applications will target .NET and will execute wherever .NET is implemented instead of targeting a Hardware/OS combination. The components that make up the .NET platform are collectively called the .NET Framework. The .NET Framework is a managed, type-safe environment for developing and executing applications. The .NET Framework manages all aspects of program execution, like, allocation of memory for the storage of data and instructions, granting and denying permissions to the application, managing execution of the application and reallocation of memory for resources that are not needed. The .NET Framework is designed for cross-language compatibility. Cross-language compatibility means, an application written in Visual Basic .NET may reference a DLL file written in C# (CSharp). A Visual Basic .NET class might be derived from a C# class or vice versa.

The .NET Framework consists of two main components:

• Common Language Runtime (CLR)

• Class Libraries The advantage of C# includes

1. Powerful Windows-based Applications

2. Building Web-based Applications

3. Simplified Deployment

• Powerful, Flexible, Simplified Data Access

• Improved Coding

• Direct Access to the Platform

• Full Object-Oriented Constructs

• XML Web Services

• COM Interoperability

The ECMA standard lists these design goals for C#:

• C# language is intended to be a simple, modern, general-purpose, object-oriented programming language.

• The language, and implementations thereof, should provide support for software engineering principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection. Software robustness, durability, and programmer productivity are important.

• The language is intended for use in developing software components suitable for deployment in distributed environments.

• Source code portability is very important, as is programmer portability, especially for those programmers already familiar with C and C++.

• Support for internationalization is very important.

• C# is intended to be suitable for writing applications for both hosted and embedded systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions.

• Although C# applications are intended to be economical with regard to memory and processing power requirements, the language was not intended to compete directly on performance and size with C or assembly language.

### 6.1.2 Features

• It has no global variables or functions. All methods and members must be declared within classes. Static members of public classes can substitute for global variables and functions.

• Local variables cannot shadow variables of the enclosing block, unlike C and C++. Variable shadowing is often considered confusing by C++ texts.

• C# supports a strict Boolean data type, bool. Statements that take conditions, such as while and if, require an expression of a type that implements the true operator, such as the Boolean type. While C++ also has a Boolean type, it can be freely converted to and from integers, and expressions such as if(a) require only that a is convertible to bool, allowing a to be an int, or a pointer. C# disallows this "integer meaning true or false" approach, on the grounds that forcing programmers to use expressions that return exactly bool can prevent certain types of common programming mistakes in C or C++ such as if (a = b) (use of assignment = instead of equality ==).

• In C#, memory address pointers can only be used within blocks specifically marked as unsafe, and programs with unsafe code need appropriate permissions to run. Most object access is done through safe object references, which always either point to a "live" object or have the well-defined null value; it is impossible to obtain a reference to a "dead" object (one that has been garbage collected), or to a random block of memory. An unsafe pointer can point to an instance of a value-type, array, string, or a block of memory allocated on a stack. Code that is not marked as unsafe can still store and manipulate pointers through the System.IntPtr type, but it cannot dereference them.

• Managed memory cannot be explicitly freed; instead, it is automatically garbage collected. Garbage collection addresses the problem of memory leaks by freeing the programmer of responsibility for releasing memory that is no longer needed.

• In addition to the try...catch construct to handle exceptions, C# has a try...finally construct to guarantee execution of the code in the finally block.

• Multiple inheritance is not supported, although a class can implement any number of interfaces. This was a design decision by the language's lead architect to avoid complication and simplify architectural requirements throughout CLI.

• C#, like C++, but unlike Java, supports operator overloading.

• C# is more type safe than C++. The only implicit conversions by default are those that are considered safe, such as widening of integers. This is enforced at compile-time, during JIT, and, in some cases, at runtime. No implicit conversions occur between Booleans and integers, nor between enumeration members and integers (except for literal 0, which can be implicitly converted to any enumerated type). Any user-defined conversion must be explicitly marked as explicit or implicit, unlike C++ copy constructors and conversion operators, which are both implicit by default. Starting with version 4.0, C# supports a "dynamic" data type that enforces type checking at runtime only.

• Enumeration members are placed in their own scope.

• C# provides properties as syntactic sugar for a common pattern in which a pair of methods, accessor (getter) and mutator (setter) encapsulate operations on a single attribute of a class.

• Checked exceptions are not present in C# (in contrast to Java). This has been a conscious decision based on the issues of scalability and versionability. [28]

• Though primarily an imperative language, since C# 3.0 it supports functional programming techniques through first-class function objects and lambda expressions.

### 6.2. FUTURE WORK

As a part of security, the pixels of the cover image are filtered both according to their position and the threshold limit. Because of this, the space availability of data insertion could become very less. Therefore, the embedding information should be small for successful embedding. New ideas could be developed on increasing the space availability in the cover image to insert as much data as possible**.** And also this project has an assumption that is both the sender and receiver must have shared some secret information before imprisonment. Pure steganography means that there is none prior information shared by two communication parties. The future enhancements of this project ***are to make it pure steganography application***.

## APPENDICES

### Steganography vs Cryptography

Basically, the purpose of cryptography and steganography is to provide secret communication. However, steganography is not the same as cryptography. Cryptography hides the contents of a secrete message from a malicious people, whereas steganography even conceals the existence of the message. In cryptography, the system is broken when the attacker can read the secret message. Breaking a steganography system need the attacker to detect that steganography has been used.

It is possible to combine the techniques by encrypting message using cryptography and then hiding the encrypted message using steganography. The resulting stego-image can be transmitted without revealing that secret information is being exchanged.

**Appendix 1 Glossary of Abbreviations and Classified Business Terms**

|  |  |  |
| --- | --- | --- |
| **Ref** | **Abbreviation** | **Full IT Application Name** |
| 1 | PC | Personal Computer (desktop or laptop) |
| 2 | PDA | Personal Digital Assistant; a portable computing device. |
| 3 | Windows XP | A Windows-based operating system |
|  |  |  |