**A Project Report on**

**Extended Steganography**

**Submitted to the Dept. of Information Technology, SNIST**

**In the partial fulfillment of the academic requirements for the award of**

**B. Tech (IT) under JNTU**

**By**

**Mr. Y. Sai Krishna (11311A1274)**

**Mr. S.Vivek Roshan (11311A12A8)**

**Mr. K. S. S. Murthy (12315A1222)**

**Under the guidance of**

**Mr. M. Naga Raju**

**Assistant professor**

**SNIST**

**Internal Guide**

****

**Department of Information Technology**

**School of Computer Science and Informatics**

**Sreenidhi Institute of Science and Technology**

**Yamnampet, Ghatkesar Mandal, R.R Dist., Hyderabad – 501 301**

**Affiliated to**

**Jawaharlal Nehru Technological University**

**Hyderabad-500 085**

**2014-15**

**Department of Information Technology**

**School of Computer Science and Informatics**

**Sreenidhi Institute of Science and Technology**

****

**CERTIFICATE**

This is to certify that the project report on “**Extended Steganography”** is a bonafide work carried out by **Mr. S.Vivek Roshan (11311A12A8), Mr. K. S. S. Murthy (12315A1222), Mr. Y. Sai Krishna (11311A1274)** respectively in the partial fulfillment for the award of B.Tech degree in Information Technology, Sreenidhi Institute of Science and Technology, Hyderabad, affiliated to Jawaharlal Nehru Technological University, Hyderabad under our guidance and supervision.

The results embodied in the project work have not been submitted to any other University or Institute for the award of any degree or diploma.

**Internal Guide Head of the Department**

**Mr. M. Naga Raju**  **Prof. V.S.S.S. Balaram**

**Acknowledgements**

We would like to express our immense gratitude and thanks to **Dr. V.S.S.S. Balaram**, Professor in Information Technology for his guidance, valuable suggestions and encouragement in completing the Mini-Project work within the stipulated time.

We would like to express our sincere thanks to **Dr.P. Narasimha Reddy**, Executive Director, **Dr.K. Sumanth**, Principal, **Dr.V. Umakanta Sastry**, Director, School of Computer Science and Informatics, **Dr. V.S.S.S. Balaram**, Professor & Head of the Department of Information Technology, **Mr. K. Kranthi Kumar**, Associate Professor & Mini-Project Work Coordinator of the Department of Information Technology, **Sreenidhi Institute of Science and Technology (An Autonomous Institution), Hyderabad** for permitting us to do our Mini-Project work.

Finally, we would also like to thank the people who have directly or indirectly helped us and parents and friends for their cooperation in completing the Mini-Project work.

**Mr. Y. Sai Krishna (11311A1274)**

**Mr. S.Vivek Roshan (11311A12A8)**

**Mr. K. S. S. Murthy (12315A1222)**

**Contents**

|  |  |  |
| --- | --- | --- |
| **Sl.**  **No.** | **Details** | **Page**  **No.** |
| 1. | Abstract | 1 |
| 2. | Introduction | 2 |
| 3. | Objectives | 4 |
| 4.  4.1. | Existing System  Proposed System | 5  5 |
| 5.  5.1.  5.2. | System Requirements  Software Requirements  Hardware Requirements | 6  6  6 |
| 6.  6.1.  6.2. | Technique Used  How Steganography Works?  How Cryptography Works? | 7  7  8 |
| 7.  7.1.  7.2.  7.3. | Modules  Sender Side  Receiver Side  Interface module | 9  9  10  10 |
| 8.  8.1.  8.2. | Java Language  Features of Java Language  About Swings | 11  11  12 |
| 9.  9.1. | Steganography vs. Cryptography  Steganography vs. Watermarking | 14  14 |
| 10.  10.1.  10.2. | Introduction to the UML  An Overview of the UML  A Conceptual Model of the UML | 16  16  18 |
| 11.  11.1.  11.2.  11.3. | UML Diagrams  Use Case Diagram  Sequence Diagram  Activity Diagram | 20  20  23  26 |
| 12. | Input/ Screenshots | 28 |
| 13.  13.1. | Test Cases  Steps for Manual Testing | 35  35 |
| 14.  14.1. | Conclusions and Scope for Future Enhancements  Limitations | 37  37 |
| 15. | References | 38 |

**1. ABSTRACT**

One of the reasons that intruders can be successful is the most of the information they acquire from a system is in a form that they can read and comprehend. Intruders may decode it,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 the art and science of hiding the important messages, text, and confidential information in such a way that no one apart from the intended recipient knows the existence of the message; this is in contrast to cryptography, where the existence of the message itself is not disguised, but the content is obscured. The advantage of steganography over cryptography alone is that, no one would able to know the truth that the information is hidden inside the image. Steganography is done using Least Significant Bit (LSB) algorithm, Where the Least Significant Bit in the image pixels are used to store the text which we wanted to hide.

Extended Steganography is the project where the content of the text is first changed using cryptographic methods and then the encrypted text is made to hide inside an image. With the help of steganography, only the intended recipient would know the existence of the message in the image. With the help of cryptography, the data is converted to an obscured encrypted text. Here, in this project we are encrypting the data which is followed by the data hiding. So, even when an undesired person can recover the message from the image, the text is encrypted providing the double security. The project initially ask for the image and the text to be encoded and to hide. When encode button is clicked, the text is encrypted and then hidden inside the image and a new image is formed, not replacing the original one. The project is done using Java and the GUI required is done using Java Swings.

**2. Introduction**

One of the reasons that intruders can be successful is the 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.

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.

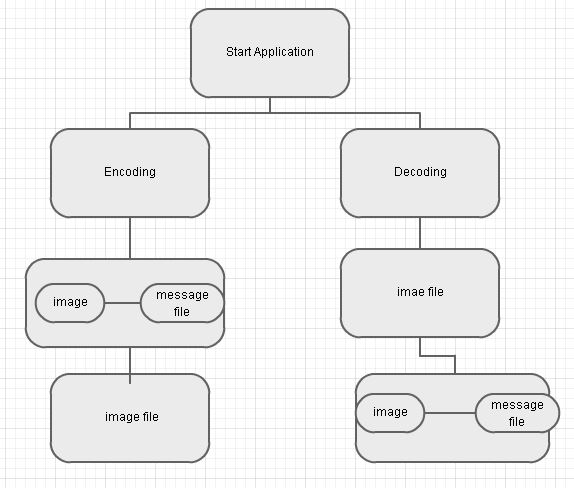
Fingerprint, the owner of the data set embeds a serial number that uniquely identifies the user of the data set. This adds to copyright information to makes it possible to trace any unauthorized used of the data set back to the user.

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.

Encryption is the process of encoding a message in such a way as to hide its contents. Modern Cryptography includes several secure algorithms for encrypting and decrypting messages. They are all based on the use of secrets called keys. A cryptographic key is a parameter used in an encryption algorithm in such a way that the encryption cannot be reversed without the knowledge of the key.

Extended Steganography is the project which is the combination of the Steganography and Cryptography where, the confidential information is first encrypted using a key and now this encrypted information is encoded in to the image using the methods of the Steganography.

The message can also be encrypted before it is hidden inside a cover message. This provides a double layer of protection. To begin with, encryption may make the existence of the message even more difficult to detect, due to the fact that some encryption techniques cause the patterns of the characters in the encrypted version to be more random than in the original version. In addition, even if the existence of the encrypted message is detected, it is unlikely that an eavesdropper will be able to read the message.



**3. Objectives**

The project has the following objectives:

1. To create a tool that can be used to hide data inside a color image.
2. The tool should be easy to use, and should use a graphical user interface.
3. The tool should work cross-platform.
4. The tool should effectively hide a message using an image degradation approach, and should be able to retrieve this message afterwards.
5. The tool should take into account the original content, to theoretically more effectively hide the message.
6. The tool should be able to provide some information as to the effectiveness of the hiding i.e. it should be able to evaluate the degradation of an image. The analysis used will consist of existing watermarking measures, re- implemented for this tool.
7. The tool should be able to encrypt the message before embedding it.
8. The technique should fall under the category of Encrypted cum Steganography -where without the key is first used to encrypt the secret text and then hide the message inside an image.

**4. Existing System:**

* Due to the increase in the cyber crimes, the field of communication has been facing many serious threats.
* One of the conventional method for ensuring the threat free communication is cryptography
* In cryptography, the message or the text is made obscure, in such a way that it is not understood by any others except the intended recipient.
* But, in the present Scenario, with the rapid increase in the development, the ciphered text is also being deciphered.
* This type of cyber crimes can be ceased, if we hide the ciphered text in an other information(such as an image ).
* So, by hiding the ciphered text in an image, the opponent would never know the existence of the confidential data inside it.

**4.1. Proposed System:**

* The System under proposal is “Extended Steganography”.
* Image Steganography is the process of hiding the information inside an image, so that the confidential information is not exposed to risk
* With the help of Steganography, important messages, confidential data can be sent, without exposing the data to the opponent.
* It has many applications, For Ex: The major of the army sending the message troop “to attack”.
* We can encrypt it, hide the data and can send it over the internet safely.

**5. Specifications:**

**5.1. Software Requirements:**

* OPERATING SYSTEM: Windows
* Java version above 1.5.0

**5.2. Hardware Requirements:**

* RAM -1GB
* HARD DISK -80GB
* PROCESSOR -Intel pentium4

**6. Technique used**

Steganographic Techniques used is oncealing data within encrypted data. The data to be concealed is first encrypted before being used to overwrite part of a much larger block of encrypted data. This technique works most effectively where the decrypted version of data being overwritten has no special meaning or use. Each pixel typically has three numbers associated with it, one each for red, green, and blue intensities, and these values often range from 0-255. Each number is stored as eight bits (zeros and ones), with a one worth 128 in the most significant bit (on the left), then 64, 32, 16, 8, 4, 2, and a one in the least significant bit (on the right) worth just 1.

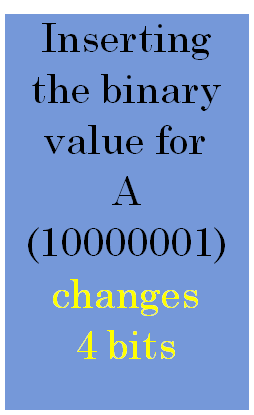
**6.1. How Steganography Works?**

Steganography strips less important information from digital content and injects hidden data in its place. This is done over the spectrum of the entire image. Here's one way it could be implemented:

The following sequence of 24 bits represents a single pixel in an image. Its 3 bytes of color information provide a total of 256 different values for each color (red, green and blue) and thus can represent a total of 16.7 million colors.

* The Method of Steganography, is achieved by LSB algorithm.
* LSB algorithm is the least significant bit algorithm, where the least significant bits of the bytes for the pixels are used to store the information.
* 8 bits make up a byte, and the right most bit of the byte is called the least significant bit.
* As the name indicates, it has the least significance and have a value of 2^0 i.e.. 1.
* Hence, embedding the text in these LSB’s would not effect the image .
* An image contains many pixels according to its resolution.
* Each pixel is associated with its RGB values, where each of it is a 8 bit value. So, each pixel is associated with 3 bytes.
* So we can embed 3 bits of our information, in one pixel.

**An LSB Example:**



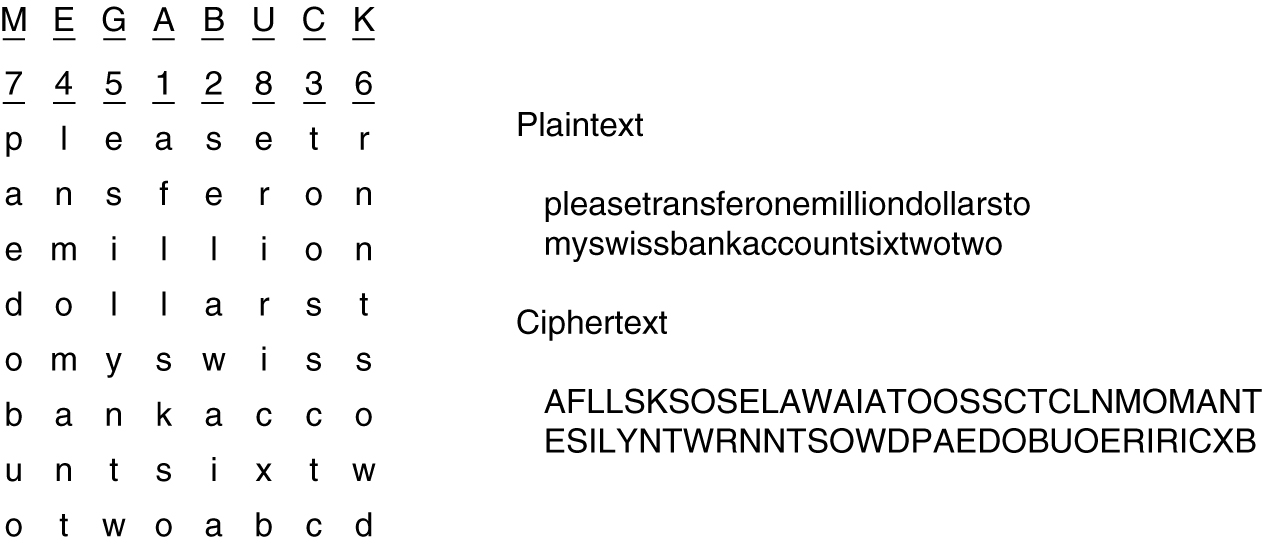
* 00100111 11101001 11001000
* 00100111 11001000 11101001
* 11001000 00100111 11101011



* 00100111 11101000 11001000
* 00100110 11001000 11101000
* 11001000 00100111 11101011

**6.2. How Cryptography Works?**

**Transposition Cipher**: A transposition cipher is keyed by a phrase such as “MEGABUCK”. The letter in the key indicated the order of columns to be output. Plaintext horizontally read in, cipher text read out column by column.

****

7. MODULES

The project consists of mainly four modules encryption, hide message, retrieve message and decryption. Encryption and hide message form the modules at the sender side. retrieve message and decryption form the modules at the receiver side

**7.1. Sender Side**

**7.1.1. Encryption (Cryptographic Module)**

Encryption includes a message or a file encrypting. Encryption involves converting the message to be hidden into a cipher text. Encryption can be done by passing a secret key.Secret key can be used for encryption of the message to be hidden.It provides security by converting it into a cipher text,which will be difficult for hackers to decrypt. Moreover if the message is password protected,then while retrieving message,the retriever has to enter the correct password for viewing the message.

**7.1.2. Hide Message (Steganographic Module)**

Hiding message is the most important module of steganography.It involves embedding the message into the cover text. Each pixel typically has three numbers associated with it, one each for red, green, and blue intensities, and these values often range from 0-255.In order to hide the message,data is first converted into byte format and stored in a byte array.The message is then encrypted and then embed each bits into the LSB position of each pixel position. The least significant (rightmost) bit of each 8-bit byte has been co-opted to hide a text message.

**7.2. Receiver Side**

**7.2.1. Retrieve information (Steganographic module)**

It involves retrieving the embed message from the file independent of the file format. Once the message has been retrieved it has to be converted into original message or file. This can be done by reading the embedded data from the master file. The read data will be in the bytes format. This message has to be converted into the suitable output file format.

**7.2.2 Decryption (Cryptographic Module)**

Decryption includes a message or a file decrypting. Decryption involves converting the cipher text into decrypted format. Decryption can be done by passing a secret key. Secret key can be used for decryption of the message that is hidden. It provides security by converting the cipher text, into the original data message or file . Moreover if the message is password protected, then while retrieving message, the retriever has to enter the correct password for viewing the message.

**7.3. User Interface Module:**

This module works with creating a Graphical User Interface(GUI), in order to have a user friendly interface. A user friendly graphical user interface can be created with the help of java swings. The applications steganography and cryptography both have the different interfaces with required buttons, labels etc. to achieve the process of encryption and hiding the information.

8. Java

Java was developed at Sun Microsystems. Work on Java initially began with the goal of creating a platform-independent language and OS for consumer electronics. The original intent was to use C++, but as work progressed in this direction, developers identified that creating their own language would serve them better. The effort towards consumer electronics led the Java team, then known as First Person Inc., towards developing h/w and s/w for the delivery of video-on-demand with Time Warner.

Today Java is both a programming language and an environment for executing programs written in Java Language. Unlike traditional compilers, which convert source code into machine level instructions, the Java compiler translates java source code into instructions that are interpreted by the runtime Java Virtual Machine. So unlike languages like C and C++, on which Java is based, Java is an interpreted language.

Java is the first programming language designed from ground up with network programming in mind. The core API for Java includes classes and interfaces that provide uniform access to a diverse set of network protocols. As the Internet and network programming have evolved, Java has maintained its cadence. New APIs and toolkits have expanded the available options for the Java network programmer.

**8.1. Features of java:**

In one of their early papers about the language, Sun described Java as follows: Java: A simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, and dynamic language. Sun acknowledges that this is quite a string of buzzwords, but the fact is that, for the most part, they aptly describe the language. In order to understand why Java is so interesting, let's take a look at the language features behind the buzzwords.

**Platform Dependent:**

The concept of Write-once-run-anywhere (known as the platform independent) is one of the important key feature of java language that makes the java the most powerful language. The programs written on one platform can run any platform provided the platform must have the JVM.

**Simple:**

There are various features that make the java as a simple language. Programs are very easy to write and debug because java does not use the pointers explicitly.

**Object Oriented:**

To be an Object Oriented language, any language must follow at least the four characteristics.

**Inheritance:**

It is the process if creating the new classes and using the behavior of the existing classes by extending them just to reuse the existing code and adding the additional features as needed.

**Encapsulation:**

The wrapping up of data and functions into a single unit ( called class) is known

as **encapsulation**. Data encapsulation is the most striking features of a class.

**Polymorphism:**

It allows the single method to perform different actions based on the parameters.

**Dynamic Binding:**

When a method is called within a program, it associated with the program at run time rather than at compile time is called dynamic binding.

**Portable:**

The feature Write-once-run-anywhere makes the java language portable provided that the system must have interpreter for the JVM. Java also has the standard data size irrespective of operating system or the processor. These features make java as a portable language.

**8.2. About Swings:**

Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit (AWT). Swing provides a native look and feel that emulates the look and feel of several platforms, and alsosupports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists. Unlike AWT components, Swing components are not implemented by platform-specific code. Instead they are written entirely in Java and therefore are platform-independent. The term "lightweight" is used to describe such an element

A Java toolkit for developing graphical user interfaces (GUIs). It includes elements such as menus, toolbars and dialog boxes. Swing is written in Java and is thus platform independent, unlike the Java Abstract Window Toolkit (AWT), which provides platform-specific code. Swing also has more sophisticated interface capabilities than AWT and offers such features as tabbed panes and the ability to change images on buttons. Swing is included in the Java Foundation Classes (JFC) which is provided in the Java Developers Toolkit (JDK).

**9. 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 conceal 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.

**9.1. Steganography vs. Watermarking:**

Steganography pay attention to the degree of Invisibility while watermarking pay most of its attribute to the robustness of the message and its ability to withstand attacks of removal, such as image operations(rotation, cropping, filtering), audio operations(rerecording, filtering)in the case of images and audio files being watermarked respectively.

It is a non-questionable fact that delectability of a vessel with an introduced data (steganographic message or a watermark) is a function of the changeability function of the algorithm over the vessel.

Invisibility

Robustness

Security

That is the way the algorithm changes the vessel and the severity of such an operation determines with no doubt the delectability of the message, since delectability is a function of file characteristics deviation from the norm, embedding operation attitude and change severity of such change decides vessel file delectability.

A typical triangle of conflict is message Invisibility, Robustness, and Security. Invisibility is a measure of the in notability of the contents of the message within the vessel.

Security is sinominous to the cryptographic idea to message security, meaning inability of reconstruction of the message without the proper secret key material shared.

Robustness refers to the endurance capability of the message to survive distortion or removal attacks intact. It is often used in the watermarking field since watermarking seeks the persistence of the watermark over attacks, steganographic messages on the other hand tend to be of high sensitivity to such attacks. The more invisible the message is the less secure it is (cryptography needs space) and the less robust it is (no error checking/recovery introduced).The more robust the message is embedded the more size it requires and the more visible it is.

**10. Introduction to the UML:**

Theunified modeling language (UML) is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document the artifacts of software-intensive system. The UML was released in 1997 as a method to diagram software design. It was designed by a consortium of the best minds in object oriented analysis and design. It is by far the most exciting thing to happen to the software industry in recent years. Every other engineering discipline has a standard method of documentation. Electronic engineers have schematic diagrams, architects and mechanical engineers have blueprints and mechanical diagrams. The software industry now has UML.

**10.1. An Overview of the UML:--**

The UML is a language for

* Visualizing
* Specifying
* Constructing
* Documenting

The UML is a language for Visualizing:--

For many programmers, the distance between thinking of an implementation and then pounding it out in code is close to zero. You think it, you code it. In fact, some things are best cast directly in code. Text is a wonderfully minimal and direct way to write expressions and algorithms.

In such cases the programmer is still doing some modeling. He or she may even sketch out a few ideas on a white board or on a napkin. However some problems are exists.

The UML is a language for Specifying:--

Specifying means building models that are precise, unambiguous and complete. In particular, the UML addresses the specification of all the important analysis, design and implementation decision that must be made in developing and deploying a software intensive system.

The UML is a language for Constructing:--

The UML is not a visual programming language, but its models can be directly connected to a variety of programming languages. This means it is possible to map a model in the UML, to a programming language such as java, c++, or visual basic or even to a table in a relational databases or the persistent store of an OO databases.

The UML is a Language for Documenting:--

Healthy software Organizations produces all sorts of artifacts in addition to raw executable code. These artifacts include

* Requirements
* Architecture
* Design
* Source code
* Project plans
* Tests
* Prototypes
* Releases

Consider some of the benefits of UML:--

* 1. Your s/w system is professionally designed and documented before it is coded. You will know exactly what you are getting an advance.
  2. Since system design comes first, reusable code is easily spotted and coded with the highest efficiency. You will have lower development costs.
  3. Logic ‘holes’ can be spotted in the design drawings. Your software will behave as you expect it to. There are fewer surprises.
  4. The overall system design will dictate the way the s/w is developed. The right decisions are made before you are married to poorly written code. Again, your overall costs will be less.
  5. UML lets us see the big picture. We can develop more memory and processor efficient systems.
  6. When we come back to make modifications to your system, it is much easier to work on a system that has UML documentation. Much less ‘relearning ‘takes place. Your system maintenance costs will be lower.
  7. If you should find the need to work with another developer, the UML diagrams will allow them to get up to speed quickly in your custom system. Think of it as a schematic to a radio. How could a tech fix it without it?
  8. If we need to communicate with outside contractors or even your own programmers, it is much more efficient.

**10.2. A Conceptual Model of the UML:--**

To understand UML, you need to form a conceptual model of the languages, and this request learning there major elements, the uml’s basic building blocks, the rules that dictate how those building blocks may be put together, and some common mechanisms that apply throughout the UML.

Building blocks of the UML:--

The vocabulary of the UML encompasses three kinds of building blocks:

1. **Things.**
2. **Relations.**
3. **Diagrams.**

Things are the abstractions that are first class citizen in a model; relationships tie these things together diagrams group interesting collection of the things.

There are four kinds of things in UML

1. Structural things
2. Behavior things
3. Grouping things
4. Annotational thing

***Structural things***: Structural thingsare the nouns of UML models. These are the mostly static parts of a model, representing elements that are either conceptual or physical. In all, there are seven kinds of structural things.

First, a *Class* is a description of a set of objects that share the same attributes, operations, relationships and semantics. A class implements one or more interfaces. Graphically a class is rendered as a rectangle, usually including its name, attributes , and operations.

Second and *Interface* is a collection of operations that specify a service of a class or component. An interface there fore describes the externally visible behavior of that element. An *Interface* defines a set of operations specifications but never a set of operation implementations.

Third, Collaborationdefines an interaction and is a society of roles and other element that work together to provides some cooperative behavior that’s bigger than the sum of all the element. Therefore, collaboration have structural, as well as behavioral, dimensions. A given class might participate in several collaborations.

Fourth, a *use case* is a description of set of sequence of actions that a system performs that yields an observable result of value to particular. A use case is used to structure the behavioral things in a model, a use case is a realized by a collaboration.

Fifth, an *Active* class is a class whose objects own one or more processes or threads and therefore can initiate control activity. An active class just like a class expects that its objects represent elements whose behavior is concurrent with other elements.

Sixth, a *Component* is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces component typically represent the physical packaging of otherwise logical elements, such as classes, interfaces, and collaborations.

Seventh, a *node* is a physical element that exists at runtime and represents a computational resources generally having at least some memory and often, processing capability. A set of components may residue on a node and may also migrate from node to node.

***Behavioral things:*** Behavioral things are dynamic part of UML model. These are the verbs of a model, representing behavior overtime and space. In all, there are two primary kinds of behavioral things.

First, an *Interaction* is a behavioral that comprises a set of messages exchanged among a set of objects within a particular context to accomplish a specific purpose the behavior of a society of objects or of an individual operation may be specified with an interaction. An interaction involves a number of other elements, including messages, action sequences.

Second, a *State Machine* is a behavior specifies the sequences of states an object or an interaction goes through during its lifetime in response to events, together, with its responses those events. The behavior of an individual class or a collaboration of classes ay be specified with a state machine. A state machine involves a number of the elements, including states, transitions, event, and activities.

***Grouping Things*** : Grouping Things are organizational parts of the UML models. These are the boxes into which a model can be decomposed. In all, there is one primary kind of grouping things, namely , packages.

A ***package*** is a general-purpose mechanism for organizing elements in to group. Structural Things, Behavioral things, and even other grouping things may be placed in a package. Unlike package is purely conceptual. Packages are the basic grouping things with which you may organize a UML model. There are variations. Such as frame works, models, and subsystems.

***Annotational Things:*** Annotational things are explanatory parts of the UML models. These are the comments you may apply to describe illuminate, and remark about any elements in a model. There is one primary kind of annotational thing, called a node. A note is a simply a symbol for rendering constraints and comments attached to and element or a collection of elements.

**11. UML Diagrams**

**11.1. Use Case Diagram:**

The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose.

Because other four diagrams (activity, sequence, collaboration and Statechart) are also having the same purpose. So we will look into some specific purpose which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.

Now when the initial task is complete use case diagrams are modelled to present the outside view. So in brief, the purposes of use case diagrams can be as follows:

* Used to gather requirements of a system.
* Used to get an outside view of a system.
* Identify external and internal factors influencing the system.
* Show the interacting among the requirements are actors.

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases.

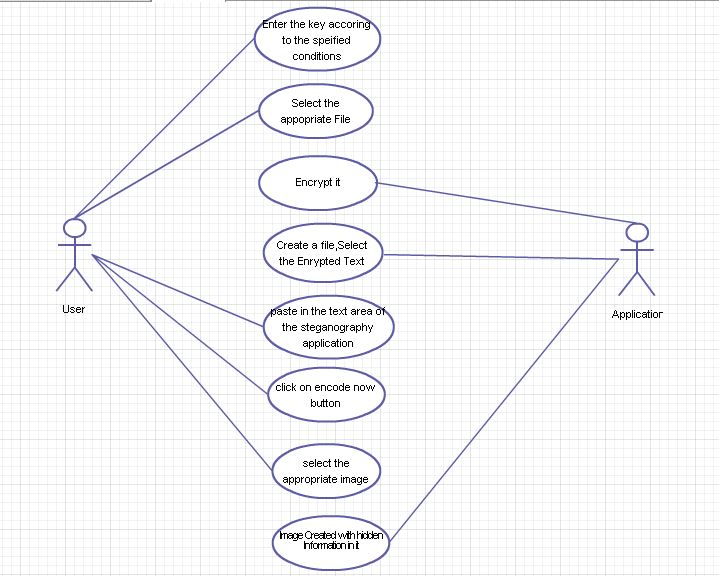
So we can say that uses cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system. The actors can be human user, some internal applications or may be some external applications. So in a brief when we are planning to draw an use case diagram we should have the following items identified.

* Functionalities to be represented as an use case
* Actors
* Relationships among the use cases and actors.

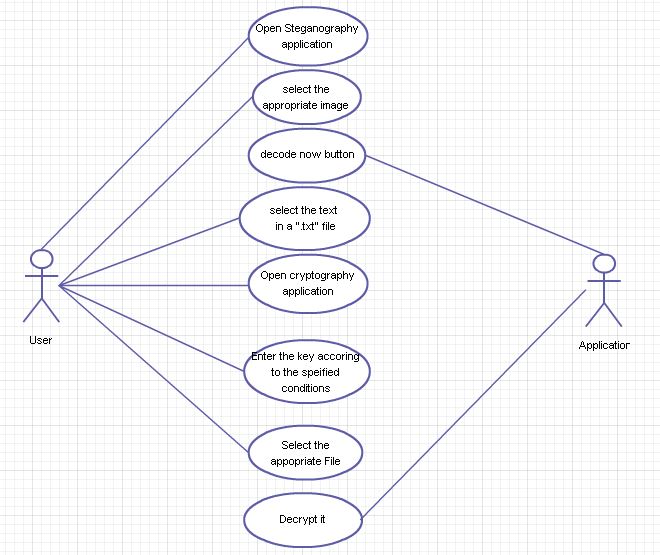
Use case diagrams are drawn to capture the functional requirements of a system. So after identifying the above items we have to follow the following guidelines to draw an efficient use case diagram.

* The name of a use case is very important. So the name should be chosen in such a way so that it can identify the functionalities performed.
* Give a suitable name for actors.
* Show relationships and dependencies clearly in the diagram.

**Use Case Diagram for Sender Side:**

****

**Use Case Diagram for Receiver Side:**

****

**11.2. Sequence Diagram:**

The purposes of interaction diagrams are to visualize the interactive behaviour of the system. Now visualizing interaction is a difficult task. So the solution is to use different types of models to capture the different aspects of the interaction.

That is why sequence and collaboration diagrams are used to capture dynamic nature but from a different angle.

So the purposes of interaction diagram can be describes as:

* To capture dynamic behaviour of a system.
* To describe the message flow in the system.
* To describe structural organization of the objects.
* To describe interaction among objects.

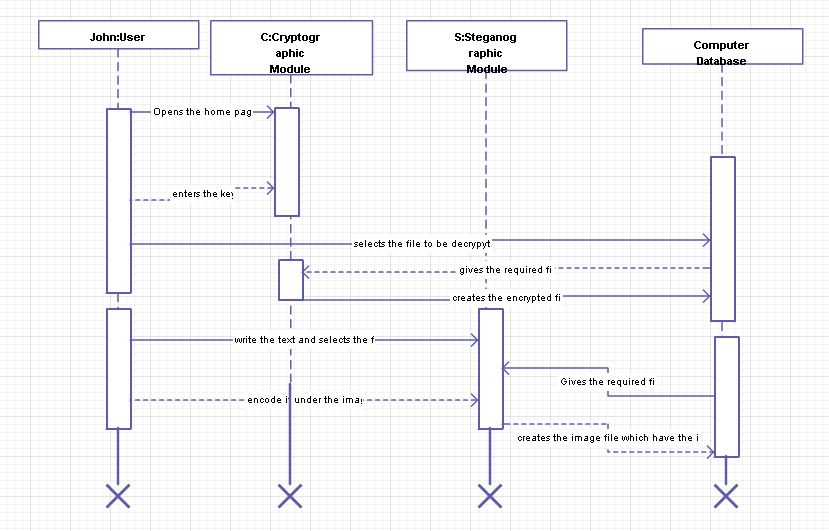
We have two types of interaction diagrams in UML. One is sequence diagram and the other is a collaboration diagram. The sequence diagram captures the time sequence of message flow from one object to another and the collaboration diagram describes the organization of objects in a system taking part in the message flow.

So the following things are to identified clearly before drawing the interaction diagram:

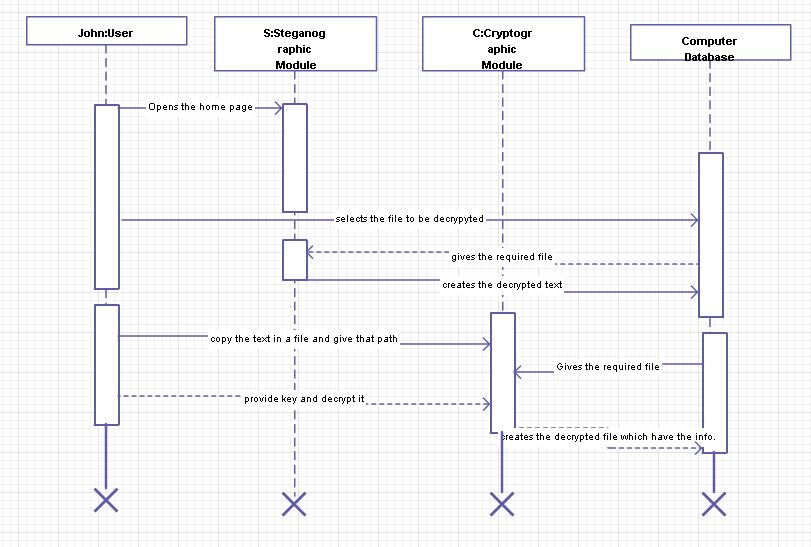
* Objects taking part in the interaction.
* Message flows among the objects.
* The sequence in which the messages are flowing.
* Object organization.

Following are two interaction diagrams modeling order management system. The first diagram is a sequence diagram and the second is a collaboration diagram.

**Sequence Diagram for Sender Side:**

****

**Sequence Diagram for Receiver Side:**

****

**11.3. Activity Diagrams**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system.

Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system.

So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc.

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

So the purposes can be described as:

* Draw the activity flow of a system.
* Describe the sequence from one activity to another.
* Describe the parallel, branched and concurrent flow of the system.

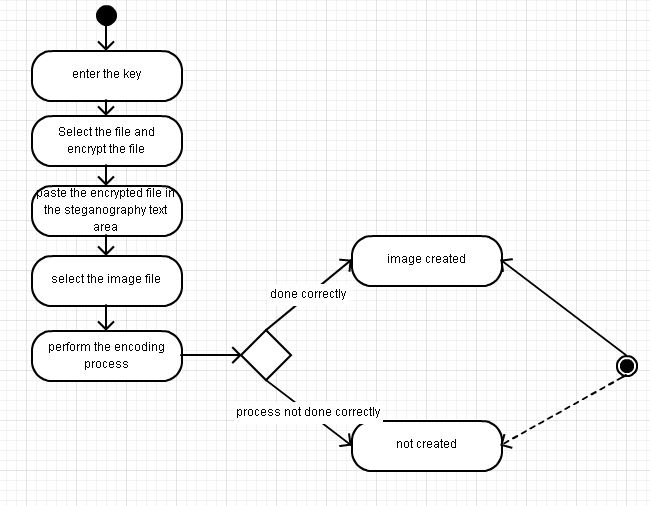
So before drawing an activity diagram we should identify the following elements:

* Activities
* Association
* Conditions
* Constraints

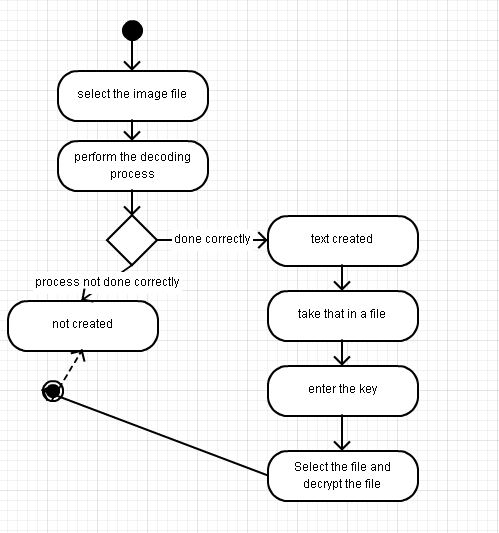
Following are the main usages of activity diagram:

* Modeling work flow by using activities.
* Modeling business requirements.
* High level understanding of the system's functionalities.
* Investigate business requirements at a later stage.

**Activity Diagram for Sender Side:**

****

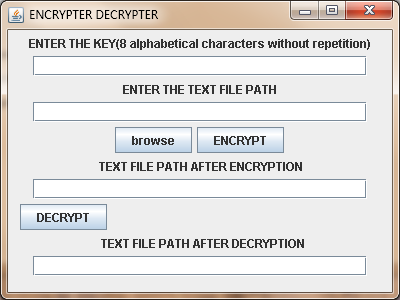
**Activity Diagram for Receiver Side:**

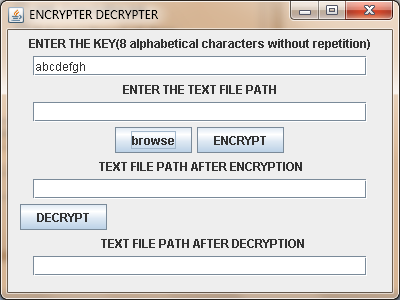
****

**12. Input/ Output Snapshots:**

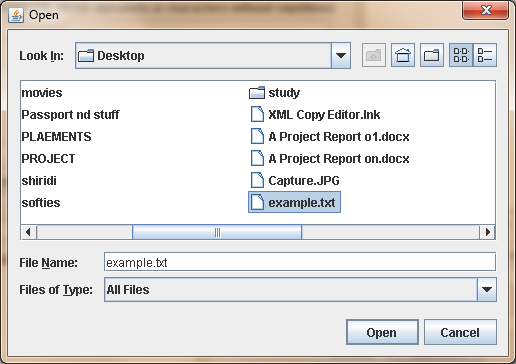
Cryptography Module Consists of the below shown Screens.

The Key is to be entered into the first label and the file is to be selected using the browse button as shown in snapshot 12.a.

****

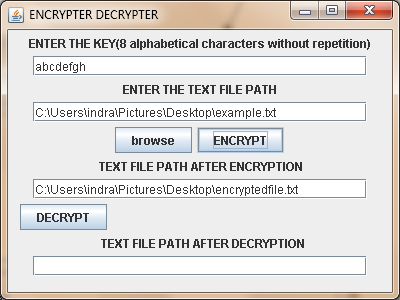


**12. a**

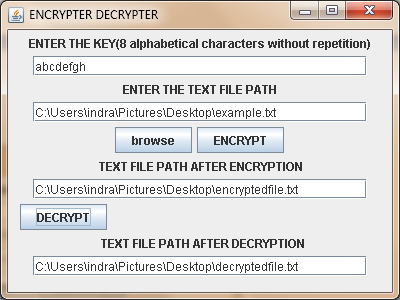
****

The Path is displayed in the second label.

By clicking the Encrypt Button, a new file is created which is the encrypted version of the given file

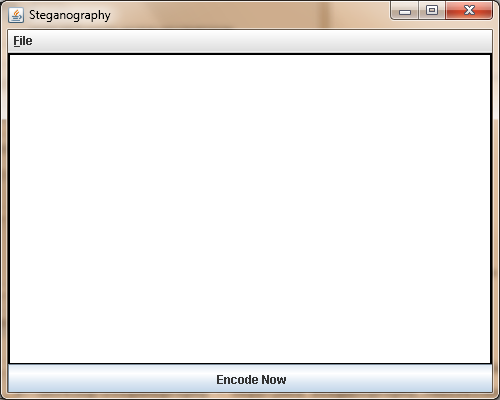
****

By clicking the decrypt Button, a new file is created which is the decrypted version of the given encrypted file..

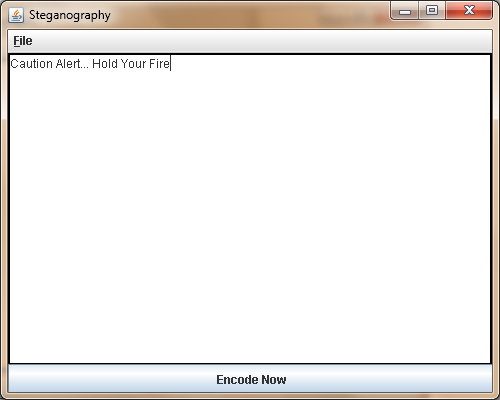
****

**Steganography Module:**

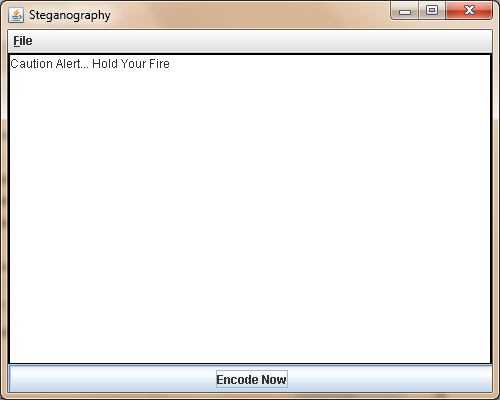
It contains the text area, which is useful in writing the text which we want to encode under an image.

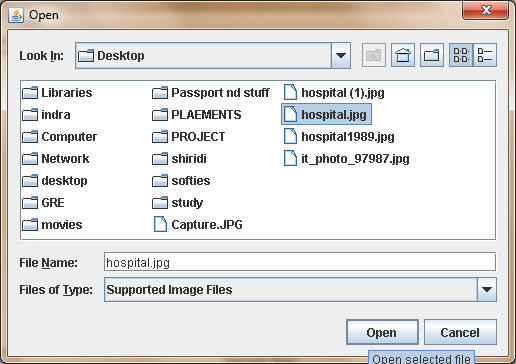
****

Give the information which you want to hide.

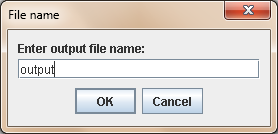
****

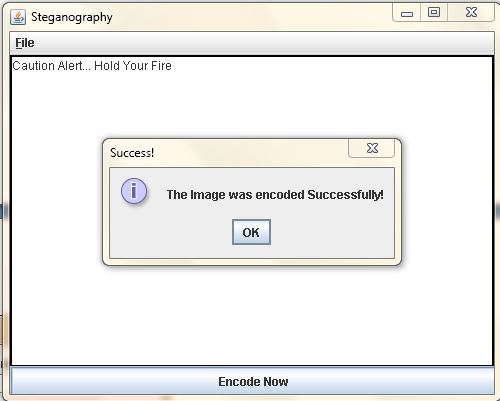
Click on Encode Now Button and select the Image File

****

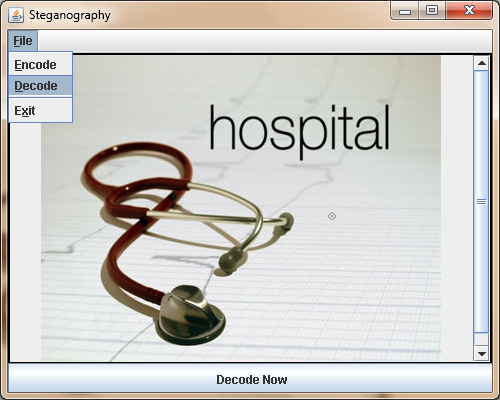
****

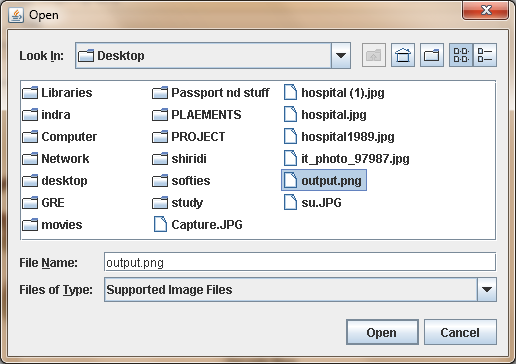
Give the name to the output File

****

****

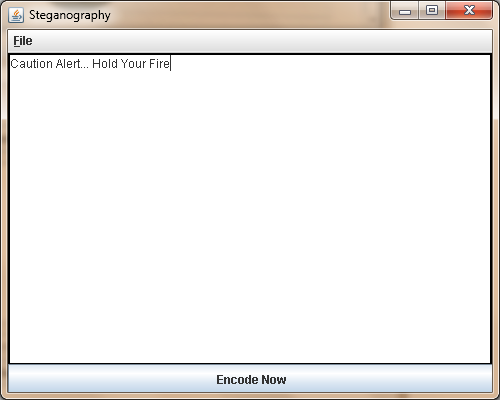
Find the encoded image and click on Decode Now button to decode.

****

****

****

The original information, which we hide under the image.

****

**13. Test Case**

A Test case is a set of conditions and variables under which a tester will determine whether a system under test identifies requirements on works correctly the process of developing test cases can also help requirement on design of an application

**Manual Testing**

Testing when carried out without any help of any tool is called manual testing. When user is required to carry out every activity related to testing manually say it as a manual testing process

**13.1. Steps for Manual Testing**

* Understand the functionality of an application
* Prepare test plan
* Write the test case and execute it
* Verify actual and expected results
* Prepare bug report
* After receiving the modified build from developers e ill go for testing and regression testing, These process are repeated until bug status is closed

**Test case description**

A test case is step by step instructions to test a specific requirements on step by step during direction to check specific functionality

**Test scenario:** a document specifying a sequence of actions for execution of test

**Test case:** one or more i/p values execution preconditions steps for execution expected values & execution values. Conditions developed for particular object or test condition

**Test Data:** Data that exist before a test is executed & that affects or is affected by component or system under test

**14.** Conclusion and Future Scope

In the area of communication revolution, information has been an inevitable component. The attraction of web services is simplicity, firewall neutrality and lack of dependency on the implementation technology at the service end.

Efficient computing capabilities are therefore utilized. The availability of the required information at the press of a button is something favourable and therefore computers are used for this purpose. Thus, one way of potential taping is achieved.

The project "Steganography" after being tested and was found to be achieving what is meant for. But this system never provides a full proof solution for all their problems in the user point of view. The system is found to be 100% error free and ready for implementation.

The system has been designed in such a way that it can be modified with very little effort when such a need arises in the future. The system has been found to work efficiently and effectively. Due to its higher user friendliness, others may use these documents as a prototype for developing similar application.

The Future Scope of the project includes, image hiding in such a way that it is not known even after exposed to any tools. One approach is hiding it, using a broad spectrum ao that the data is not stored at the start and is stored at rndom positions of the image. So even if any analysis of the image is done, it is difficult to find out that something is hidden inside an image.

**14.1. Limitations:**

* The Size of the image must be considerably big, in order to accommodate the huge data inside it.
* An image, when analyzed with different tools would expose that some changes have been done in an image, when compared to original one.
* It provides the storing of data in an unprotected mode.
* Password leakage may occur and it leads to the unauthorized access of data

**15. References:**

1. www.google.com
2. [www.microsoft.com](http://www.microsoft.com)
3. [www.dotnetspider.com](http://www.dotnetspider.com)
4. Wikipedia.org
5. www.tutorialspoint.com
6. [www.**uml**-**diagrams**.org](http://www.uml-diagrams.org)

7. [https://creately.com/app/?tempID=gc7qvpsj1&login\_type=demo#](https://creately.com/app/?tempID=gc7qvpsj1&login_type=demo).

8. http://www.hacking-tutorial.com

**Books**:

1. Network Security Essentials Applications and standards, by William Stallings.

2. Research Paper, An Overview of Image Steganography using LSB Technique.

3. Grady Booch, James Rambaiugh, Ivar Jacobson: The Unified Model.

4. Cryptography and Network Security, Third Edition, Stallings, PHI/Pearson.