A Minor Project Proposal on

**Data Security**

Using Steganography (Image processing)

Submitted in Partial Fulfillment of the Requirements for the Degree of

**Bachelors of Engineering in Software Engineering**

under Pokhara University

Submitted by:

**Sanjay Adhikari, 14731**

**Mithun Adhikari, 14748**

|  |  |
| --- | --- |
|  | **Department of Software Engineering**  **NEPAL COLLEGE OF**  **INFORMATION TECHNOLOGY**  Balkumari, Lalitpur, Nepal |

**Abstract**

In Today’s context, security of confidential data is most important. We cannot compromise when it is about the security of your private image. Everyone share private images these days. But when you need to share the image but not sure about the person whether they will leak it or not, there is nothing you can do to protect yourself. All people are seeking for more and more security. We cannot be assured that the image we send to others will be safe and trust them blindly. Though many application has been able to come up with most of the facilities of security on image, a lack of completeness is still felt.

Our project is looking forward to target the cross platform service for the services of confidential image security and information hiding mechanism. This application will be used to set a background image to existing images. this application will add the watermark over the entire image. The watermark will be added in such a fashion that the watermark will not visible clearly while viewing the image. The background image can be viewed clearly using this application only. The original image will look exactly the same, but contains other image in the background. Whatever done to the image the background image can be viewed through this application.

Keywords: Data, Security, Steganography, Steganalysis

**LIST OF TABLES**

[Table 1: Tools to be used 7](#_Toc490006170)

[Table 2: Project Task and Schedule 10](#_Toc490006171)

**LIST OF FIGURES**

[fig 1: Encoding message within Carrier image 2](#_Toc490006294)

[fig 2 Agile Methodology 5](#_Toc490006295)

[fig 3 : Agile- Feature Driven Development life cycle 6](#_Toc490006296)

[fig 4: Block diagram for Steganography 8](file:///C:\Users\Backspace\Desktop\Sixth%20sem\proposal.docx#_Toc490006297)

**TABLE OF CONTENTS**

[1 INTRODUCTION 1](#_Toc490035817)

[1.1 Problem Statement: 1](#_Toc490035818)

[1.2 Project Overview: 1](#_Toc490035819)

[1.3 Project Objectives: 2](#_Toc490035820)

[1.4 Project Scope and Limitation: 2](#_Toc490035821)

[1.5 Significance of Study 2](#_Toc490035822)

[2 LITERATURE REVIEW 4](#_Toc490035823)

[2.1 Review 4](#_Toc490035824)

[2.2 LSB Algorithm 4](#_Toc490035825)

[3 Methodology 5](#_Toc490035826)

[3.1 Software Development Life Cycle: Agile 5](#_Toc490035827)

[3.1.1 Develop an overall model 6](#_Toc490035828)

[3.1.2 Build a features list 6](#_Toc490035829)

[3.1.3 Plan by feature 6](#_Toc490035830)

[3.1.4 Design by feature 6](#_Toc490035831)

[3.1.5 Build by features 7](#_Toc490035832)

[3.2 Technologies to be used: 7](#_Toc490035833)

[3.3 Tools to be used: 7](#_Toc490035834)

[4 Technical Description 8](#_Toc490035835)

[5 Proposed Results 9](#_Toc490035836)

[6 Project Task and Time Schedule 10](#_Toc490035837)

[7 References 11](#_Toc490035838)

# INTRODUCTION

Our project is a cross platform (currently android) which service to secured the confidential (private) image. By Using the steganography, we will encode the image within another image such that no one can even guess what’s inside. This document provides the scope and context of the project to be undertaken. It also provides a schedule for the completion of the project, including a list of all the deliverables and presentations required.

This application reads the content from the device and the users can select the image on which our application process the image by adding the. Then user again selects the image which will be hidden behind the previously selected image. Also users can select the image from the device to view the hidden image behind the selected image.

## Problem Statement:

As people face the problem of getting private images leaked and don’t know who leaked it or from where it was leaked, we decided to create the project that can solve these problems.

*How can we cover a message secretly to the image?*

*How to extract covered message from stego-image?*

Using steganography, information can be hidden in carriers such as images, audio files, text files, videos and data transmissions. In this study, we proposed a new framework of an image steganography system to hide a digital text of a secret message.

## Project Overview:

This application lets you hide text in pictures so that only other users of this app can retrieve and read the hidden secret messages. Once text is hidden in an image the saved picture is still a 'picture', it will load just like any other image and appear as it did before. The image can be saved, emailed, uploaded to the web, as before, the only difference will be that it contains hidden text.

* It can be understand by the following figure.

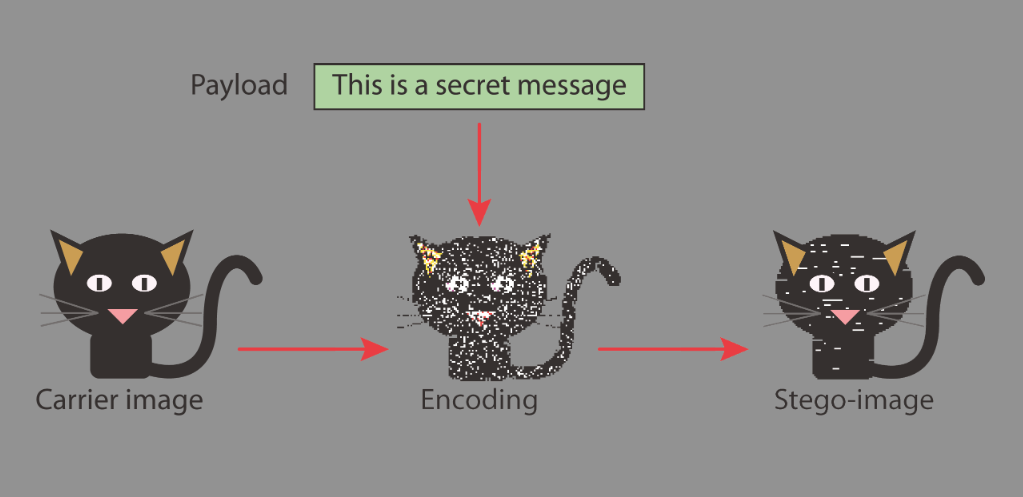


fig 1: Encoding message within Carrier image

## Project Objectives:

The objective of this project is to solve user’s problem of getting their private image from being leaked and want to trace who leaked it.

Besides transferring private image, one can just use the app to send the message to another agent embedding on the dummy image from which third party can’t even guess what’s inside that. For this, first agent uses this app for *Steganography* (adding the text in image) and second person uses this app for *Steganalysis (retrieving the text from previously generated image)*

## Project Scope and Limitation:

The scope of the project is to limit unauthorized access and provide better security during message transmission. To meet the requirements, we use the simple and basic approach of steganography. The hidden image is not visible when viewing the image outside this application, so the person viewing the image will have no idea it is actually contains a hidden image.

The limitation of this application is that if the image is stolen from either of the two persons, we cannot trace the person who has the potential of leaking the photo.

## Significance of Study

The importance of this application is that users can select the image which they want to set as a hidden background image to another image. The hidden image may contain the information about the users, the reason why they are sharing the image with that person. So the person who shares the image will have the confidence that even if the person leaks the photo, they will have a photo proof that who exactly leaked the photo. The hidden image is not visible without using this application and the foreground image looks like exactly the same as it was does not contains the hidden image.

# LITERATURE REVIEW

This section consists the literature study on the Image processing called Steganography. Our project is looking forward to define all the possible services so that there is an intelligent system of *steganography* for the required and general services

## Review

Steganography is the art and science of communicating in a way which hides the existence of the communication. Steganography plays an important role in information security. It is the art of invisible communication by concealing information inside other information.

## LSB Algorithm

Steganography is one of the most powerful techniques to conceal the existence of hidden secret data inside a cover object. Images are the most popular cover objects for Steganography and in this work image steganography is adopted. Embedding secret information inside images requires intensive computations, and therefore, designing Steganography in hardware speeds up Steganography.

* LSB (Least Significant Bit) substitution is the process of adjusting the least significant bit pixels of the carrier image.
* It is a simple approach for embedding message into the image.
* The Least Significant Bit insertion varies according to number of bits in an image.
* For an 8-bit image, the least significant bit i.e., the 8th bit of each byte of the image is changed to the bit of secret message.
* For 24-bit image, the colors of each component like RGB (red, green and blue) are changed.
* LSB is effective in using BMP images since the compression in BMP is lossless

There are mainly four steps involved in implementing LSB steganography as shown below.

1. Conversion of image to matrix
2. Encoding process
3. Conversion of matrix to image
4. Extraction process.

# Methodology

We have planned to work following these methodologies for the application of knowledge, skills, tools and techniques to a broad range of activities in order to meet the requirements of our project.

## Software Development Life Cycle: Agile

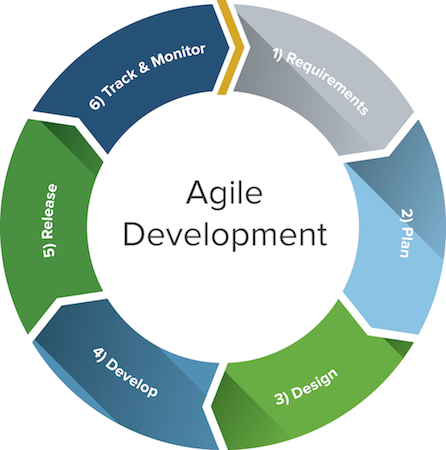
The framework we will be using for developing this project is Agile model. This is an umbrella term for several iterative and incremental software development methodologies. While each of the agile methodologies is unique in its specific approach, they all share a common vision and core values. They all fundamentally incorporate iteration and the continuous feedback that it provides to successively refine and deliver a software system. They all involve continuous planning, continuous testing, continuous integration, and other forms of continuous evolution of both the project and the software. They are all lightweight, especially compared to traditional waterfall-style processes, and inherently adaptable. What is more important about agile methods is that they all focus on empowering people to collaborate and make decisions together quickly and effectively.

fig 2 Agile Methodology

As you see in Figure below, there are five main activities in FDD that are performed iteratively.

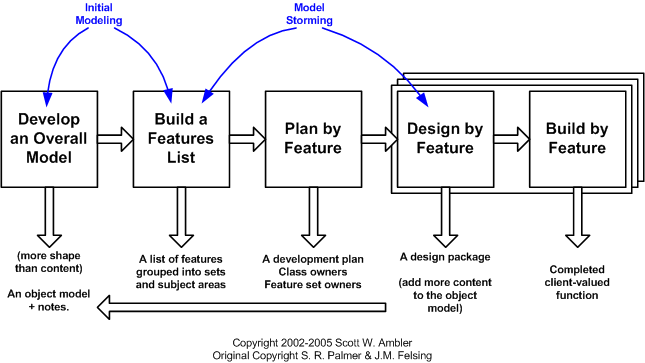


fig 3 : Agile- Feature Driven Development life cycle

### Develop an overall model

The initial result being a high-level object model and notes. At the start of a project your goal is to identify and understand the fundamentals of the domain that your system is addressing, and throughout the project you will flesh this model out to reflect what you're building.

### Build a features list

The second step is grouping them into related sets and subject areas. These first two steps map to the initial envisioning effort of AMDD. We collect the list of all the features found in our project and specialized them accordingly.

### Plan by feature

The end result being a development, the identification of class owners, and the identification of feature set owners.

### Design by feature

The majority of the effort on an FDD project, roughly 75%, is comprised of the fourth and fifth steps. In this step, we continuously model the feature in detail and build the feature iteratively until all the features are implemented.

### Build by features

In this step, the detailed design gotten from previous step are programmed, tested and packaged iteratively.

## Technologies to be used:

* React-Native, for client side app development
* PHP, backend works
* Node.js

## Tools to be used:

The tools used for documentation, designing and developing UI/UX, testing are listed below table:

|  |  |
| --- | --- |
| TOOLS | PURPOSE |
| Star UML | Design |
| Adobe Photoshop | Designing UI/UX |
| Android studio/Atom/ | Text editor for code |
| Android Emulator | Testing app real time |

Table 1: Tools to be used

# Technical Description

This application will be implemented by React-Native framework of JavaScript. Our project is based on the principle of Steganography. “The goal of steganography is to hide messages in such a way that no one apart from the intended recipient even knows that a message has been sent.”

**Pure steganography:** Pure steganography is the process of embedding the data into the object without using any private keys. This type of steganography entirely depends upon the secrecy. This type of steganography uses a cover image in which data is to be embedded, personal information to be transmitted, and encryption decryption algorithms to embed the message into image.



fig 4: Block diagram for Steganography

# Proposed Results

When our project at its final phase, it will able to provide users with the services that are essential yet not included in any other existing application. The end product will have the following end results:

* Users can encode some message within image
* Users can encrypt their private Image with Hidden message to get track of it.
* They will also have the power to promote the flow of information.
* The end product will be able to give users with a proper UI/UX so that they can enjoy service smoothly.
* Users can extract the text from the image with the help of application.

# Project Task and Time Schedule

The project schedule has been designed as per requirements and constraints involved. This project is scheduled to be completed in about 2 months. We are emphasis on *requirement analysis* and *Testing.* While carrying out a project, we will document every other thing to protect own self from being accused falsely. Documentation is evidence of a good project management. Testing and Debugging should prioritize along with the fine documentation.

|  |  |
| --- | --- |
| TASK | APPROX DURATION (in days) |
| Feasibility Study | 4 |
| Develop an overall model | 6 |
| Build a features list | 6 |
| Plan by feature | 6 |
| Design by feature | 12 (iteratively) |
| Build by feature | 29 (iteratively) |
| Develop Documentation | 55 |

Table 2: Project Task and Schedule

# References

1. Wikipedia - The Free Encyclopedia. Steganalysis. [online] 2004 May. Available at <http://en.wikipedia.org/wiki/Steganalysis>; Accessed on 26 July 2017.
2. Johnson, N. F., Jajodia, S. Steganalysis: The Investigation of Hidden Information. [online] 1998 September. Available at <http://www.jjtc.com/pub/it98a.htm>; Accessed on 25 July 2016.
3. Kessler, G. An Overview of Steganography for the Computer Forensics Examiner. [online] 2004 February. Available at <http://www.garykessler.net/library/fsc_stego.html>; Accessed on 24 July 2016.
4. Wikipedia - The Free Encyclopedia. Stegotext. [online] 2004 June. Available at <http://en.wikipedia.org/wiki/Stegotext>; Accessed on 23 July 2016.
5. React-Native official Documentation [online] Available at <https://facebook.github.io/react-native/> Accessed on 18 July 2016.