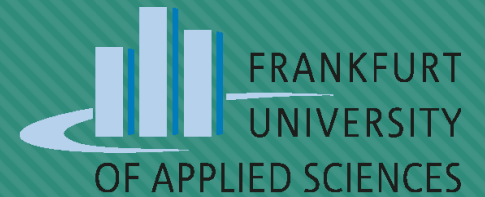


Development of A Secure Communication System Based On Steganography For Mobile Devices

A Master Thesis



17th Nov 2014

By

Dasarathan Selvaraj

Matriculation No:1028405

Advisor : Prof. Dr. Christian Baun

Co – Advisor : Prof. Dr. Matthias Wagner

Agenda

- Introduction
- Objective
- Application Feature
- Design
- Implementation
- Conclusion

Introduction

- Steganography
- Different type of steganography
- Additional security by encryption
- Development of Android app called “protectMSG” for mobile devices

Objective

- To develop an android app which implements steganography.
- It should support Android 4.0 Ice Cream Sandwich to Android 4.4 KitKat.
- Hide message into a cover file.
- Additional security for protecting the message using password.

Application Feature

- Encode Text To Image
- Encode Text To Audio
- Encode Audio To Image
- Decode Text From Image
- Decode Text From Audio
- Decode Audio From Image
- Select Image
- Capture Image
- Select audio
- Record audio
- Share
- Help

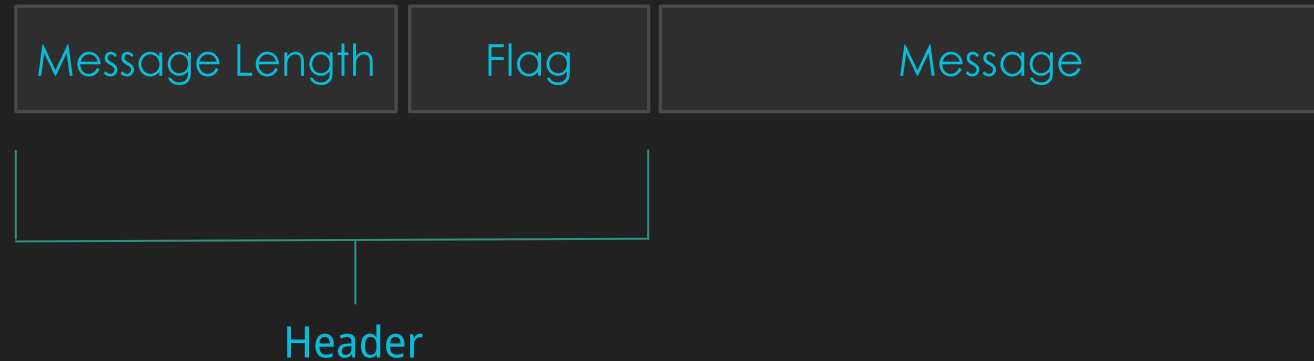
Design | LSB Technique

- Extract the bit from message byte
- Replace the LSB of Cover Byte

Message Byte	<div><div>1</div><div>0</div><div>1</div><div>0</div><div>1</div><div>1</div><div>0</div><div>0</div></div>								
	MSB				LSB				
1 st Byte	1	0	1	0	1	1	0	0	1
	0	0	1	1	0	1	0	1	0
	1	1	1	0	0	0	0	1	1
	1	0	1	0	1	1	0	0	0
Cover Bytes	1	1	0	0	0	0	0	0	1
	0	1	1	0	1	1	0	0	1
	1	0	0	0	1	1	0	1	0
	0	1	1	1	1	0	0	0	0
8 th Byte	0	1	1	1	1	0	0	0	0

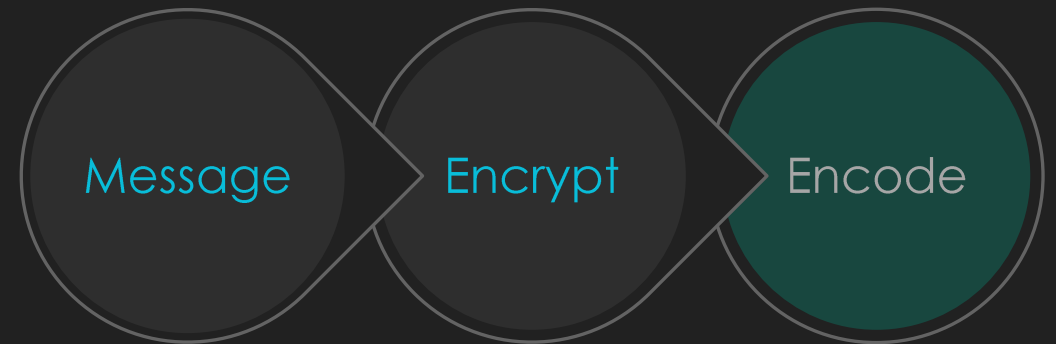
Design | Encode Data Packet

- Message Length – Size of the message to be hidden
- Flag – Used for error detection
- Message – Message to be hidden
- Frame Size
 - Message Length – 32 bits
 - Flag – 32 bits
 - Message – Variable



Implementation | Encryption

- AES Encryption
- 128 bits cipher key size
- User password is used to generate key
- Password padding for appropriate key size
- Base64 encoding scheme



Implementation

Encode Text To Image

- Enter text to hide
- Enter password for optional encryption
- Select cover Image from device storage
- Capture image using device camera to use it as a cover image.
- Share encoded image

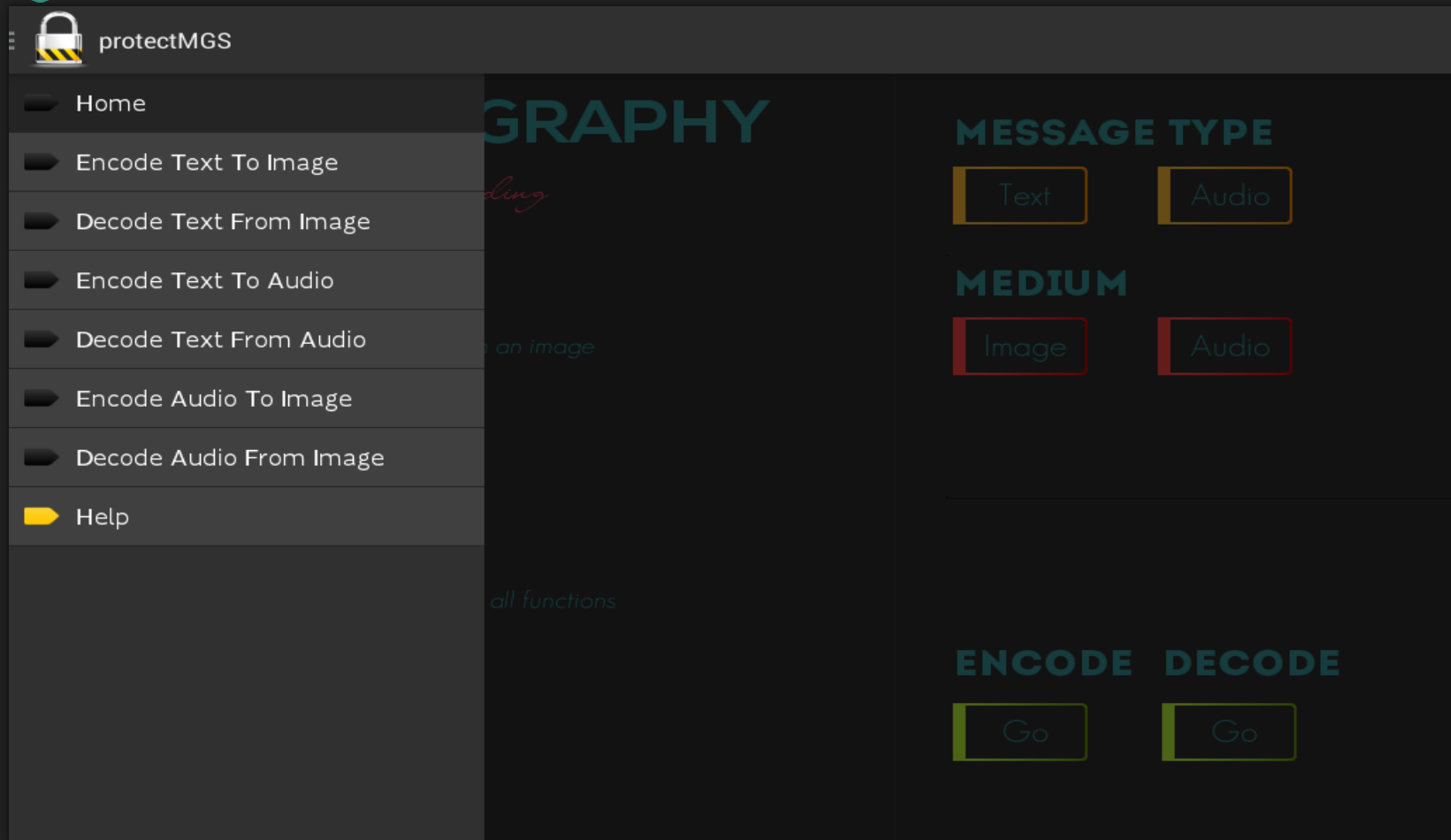
Decode Text From Image

- Select encoded image
- Enter password if the image was protected by password

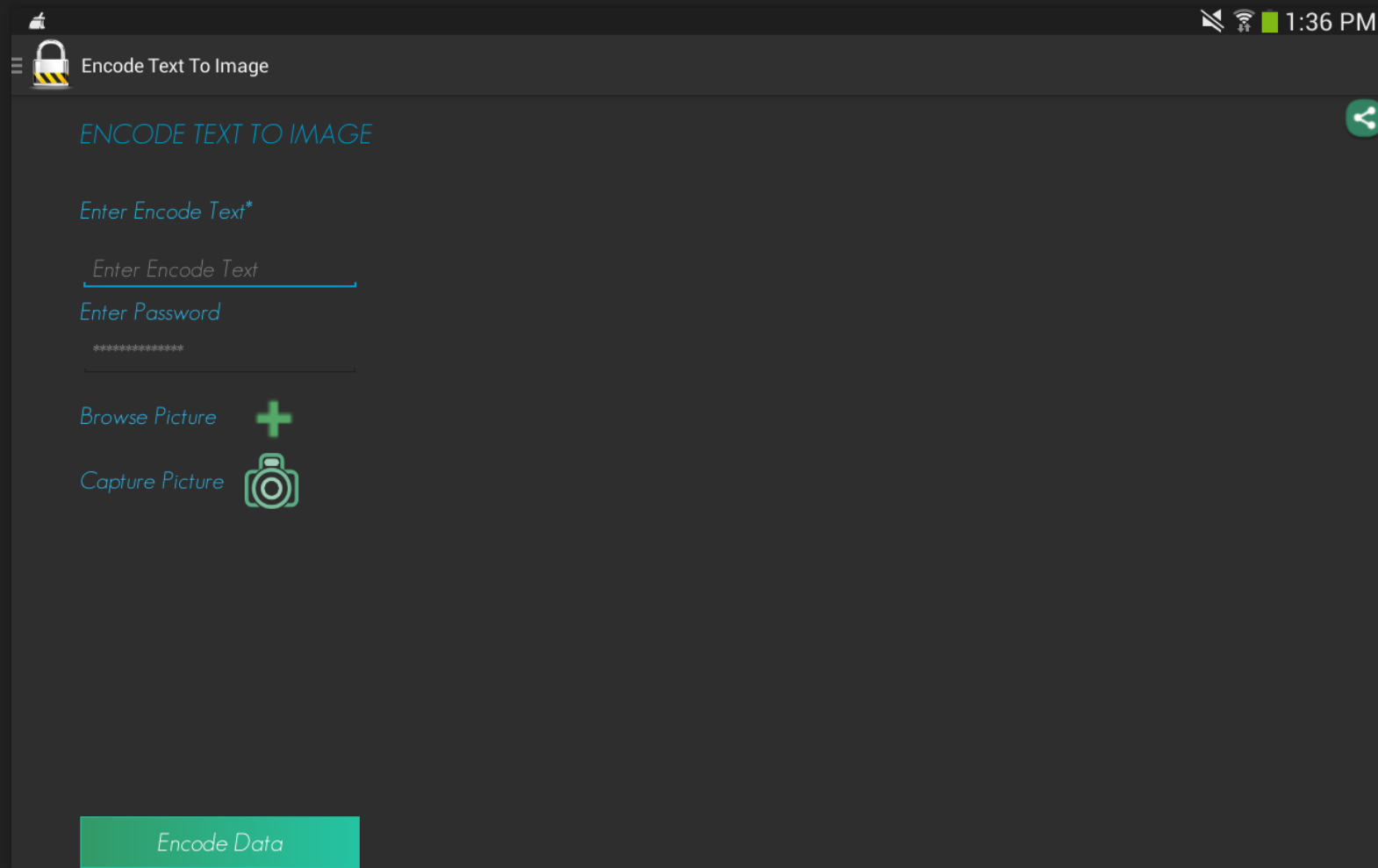
Implementation | Program Input Validation

Input	Valid Criteria
Encode Text	Not null and length between 1 to 50 characters. It can be numeric, alphabets and special characters.
Password	Null or length between 3 to 15 character. It can be numeric, alphabets and special characters.
Image	JPG, JPEG, PNG file formats are accepted.
Audio	WAV file formats is accepted.

Implementation | Home and Navigation



Implementation | Encode Text To Image



The screenshot shows an Android application interface for encoding text into an image. The app's title bar is dark gray with a yellow and black hazard icon on the left and status icons (signal, Wi-Fi, battery, and time 1:36 PM) on the right. Below the title bar, the text 'ENCODE TEXT TO IMAGE' is displayed in a light blue, monospace-style font. A green share icon is located in the top right corner of the main content area. The form consists of two text input fields: the first is labeled 'Enter Encode Text*' and contains the placeholder text 'Enter Encode Text'; the second is labeled 'Enter Password' and contains the placeholder text '*****'. Below these fields are two options: 'Browse Picture' with a green plus icon and 'Capture Picture' with a green camera icon. At the bottom of the screen is a large green button labeled 'Encode Data'.

Encode Text To Image

ENCODE TEXT TO IMAGE

Enter Encode Text*

Enter Encode Text

Enter Password

Browse Picture +

Capture Picture 📷

Encode Data



ENCODE TEXT TO IMAGE

Share file



Enter Encode Text*

Enter Encode Text

Enter Password

Browse Picture



Capture Picture



Message

Password

Choose cover object

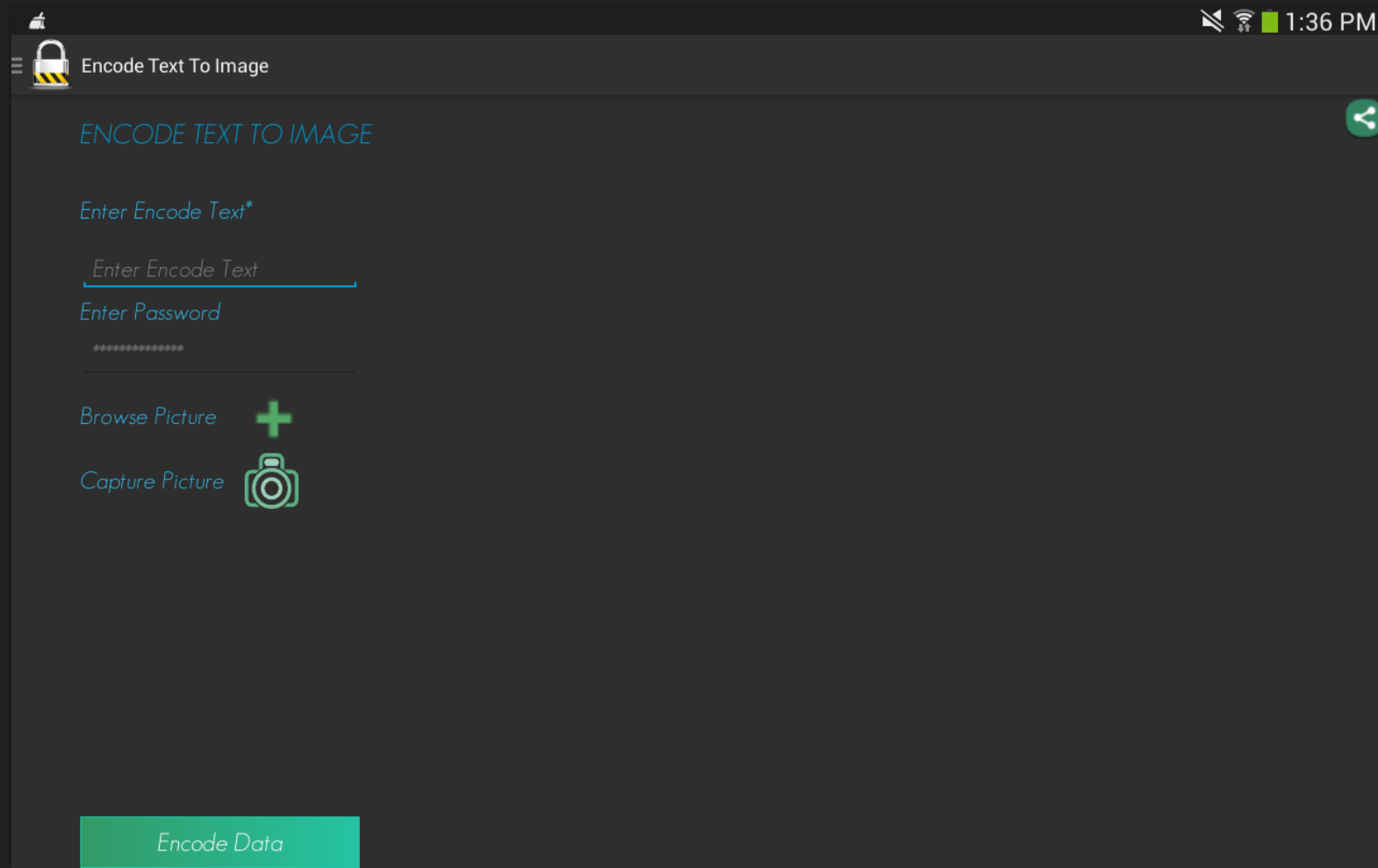
Selected cover object

Error or hidden Message

Button to start the process

Encode Data

Implementation | Encode Text To Image



The screenshot shows an Android application interface for encoding text into an image. The app's title bar is dark gray with a yellow and black hazard icon on the left and status icons (signal, Wi-Fi, battery, and time 1:36 PM) on the right. Below the title bar, the text 'ENCODE TEXT TO IMAGE' is displayed in a light blue, monospace-style font. A green share icon is located in the top right corner of the main content area. The form contains two input fields: 'Enter Encode Text*' with a light blue underline and 'Enter Password' with a light blue underline and a masked password '*****'. Below these fields are two options: 'Browse Picture' with a green plus icon and 'Capture Picture' with a green camera icon. At the bottom, there is a large green button labeled 'Encode Data'.

Encode Text To Image

ENCODE TEXT TO IMAGE

Enter Encode Text*

Enter Encode Text

Enter Password

Browse Picture +

Capture Picture 📷

Encode Data

Implementation | Supporting Features

○ Browse Audio

Browse Audio



○ Capture Picture

Capture Picture



○ Browse Picture

Browse Picture



○ Record Audio

Record Audio

Record

○ Share file



○ Selected Audio file

Audio File Selected



Audio_1413385049819.wav



Conclusion | Summary

- Implementation of LSB steganography technique in android.
- Other requirements are theoretically designed and implemented in practice as an android app named “protectMSG”
- The developed app is found to be stable and fulfills all the initial requirements.

Conclusion | Future Enhancement

- Validity for the message
- Other input file types
- Hide an image in a cover image
- Hide an audio in a cover audio

Thank You

Dasarathan Selvaraj

Master of Science,

High Integrity System,

Frankfurt University of Applied Sciences