Stegano
Final Report

By

Sean Reidy

C00227196

Institute of Technology Carlow

April 2021

Contents

Introduction

Contained in this report is a description of the project as a whole, what it is and what it is used for. There is a brief discussion of some of the different technologies which were used when completing this project. Also included is a short review of the project which will include the amount achieved and not achieved, challenges faced throughout the project and the different tests performed to verify functions within the app. A short description on what was learned and what I would change or add going forward is also included. To conclude, there will be an overall opinion made about the project and how it turned out to look and operate.

Project Description

Steganography Overview

Steganography works by replacing bits of useless or unused data with bits of different, invisible information. The purpose of steganography is covert communication to conceal a message from a third party. In comparison to this we see that cryptography relies on the art of secret writing which makes the message unreadable by a third party, but this doesn’t hide the existence of the secret message and a third party is still able to see that a message was sent from point A to point B.

Project Overview

Stegano is a tool that users can use to conceal a message within an image or audio file. Stegano also allows the user to encrypt the message before embedding it to enhance security measures. Stegano will be able to detect changes made to images by comparing MD5 hash values. The app is designed for any type of user (private or professional use).

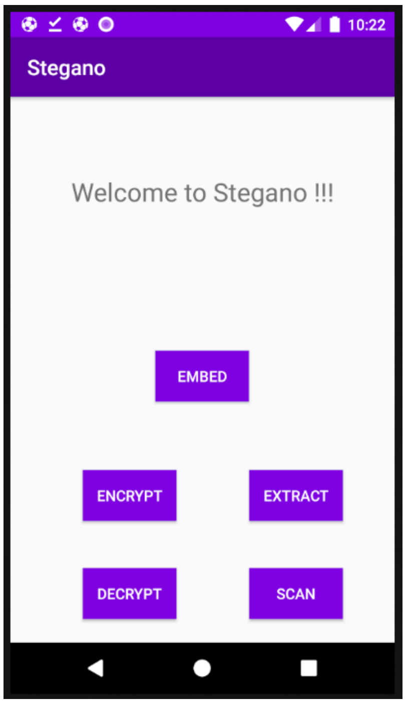
Technologies utilized

- Eclipse IDE to test various and code various functions.
- Android Studio to create the android based app
- Adobe photoshop to create the sample design in the design doc.
- Visual basic Studio to create and run small functions.

Final Design

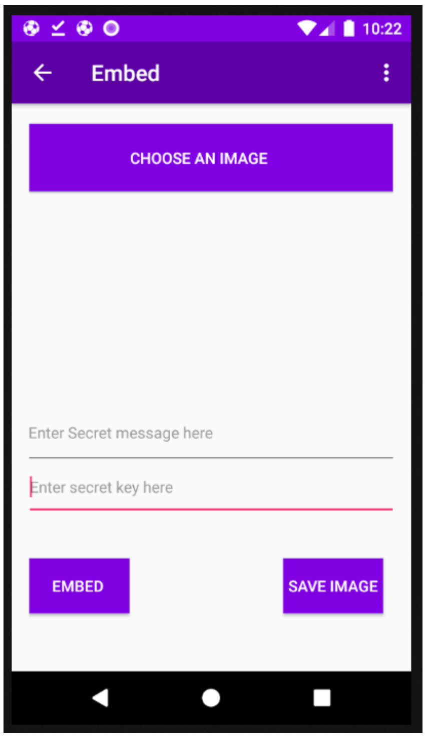
Below are screenshots of the way the application looks o both the emulator on android studio and on my android device:

Homepage:



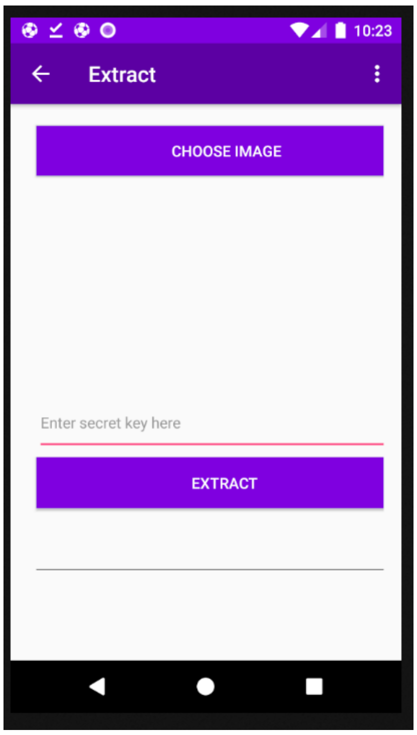
The user is presented with the homepage when they click on the app icon for Stegano. The homepage allows the user to select between the core 5 functions.

Embed Page:



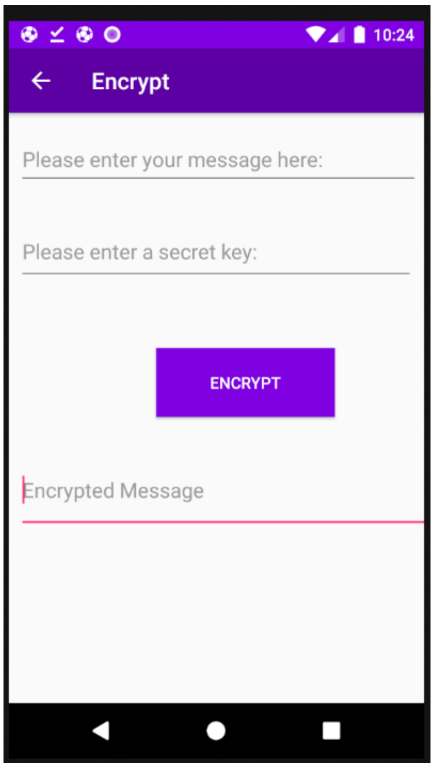
The Embed page is where the user will upload an image from their phone and embed a secret message into it. The user will need to include a secret key which will be needed in order for the extraction process. The user is forced to enter both a secret message and key and cannot proceeded with out doing so. The secret key will have to be sent through “out-of-bounds” methods. The save button saves the new stego image to the downloads folder.

Extract Page:



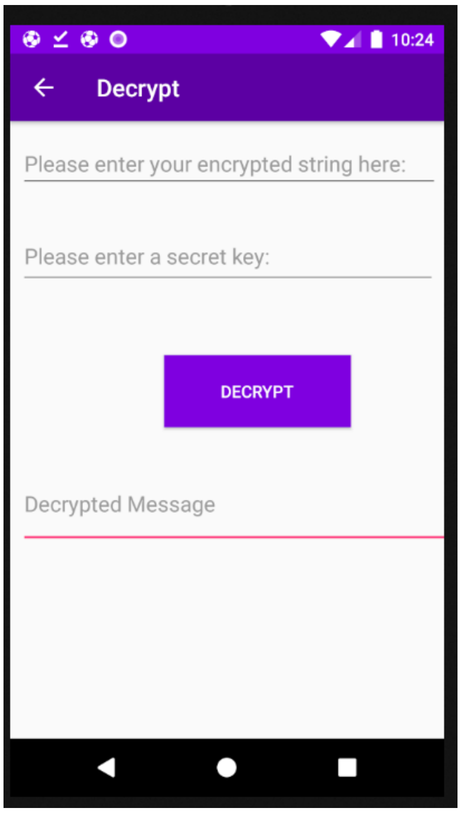
The Extract page is where the user can upload a stego image, enter the secret key and extract the secret message. If there is no secret message then the user receives an error displaying that.

Encrypt Page:



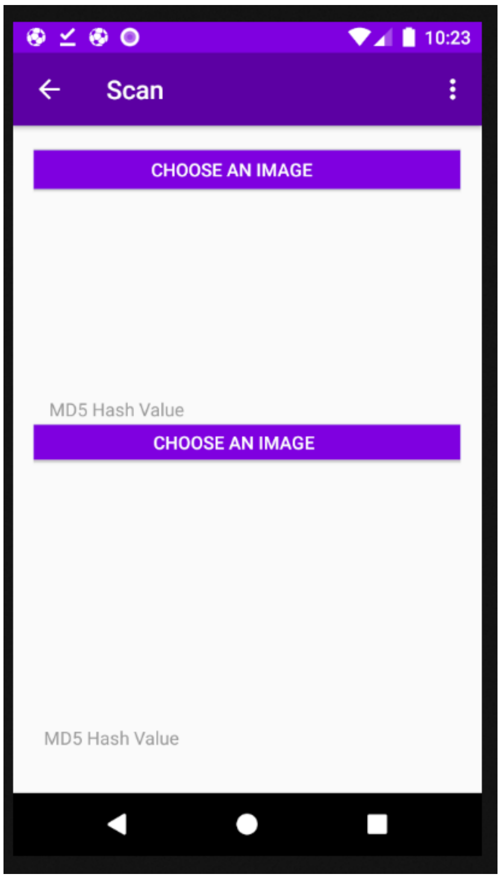
The Encrypt page is where the user will enter their secret message and a secret key so that they can encrypt it using AES base64 encoded. Once completed, the user can copy the text and use it on the embed page.

Decrypt Page:


10:24
Decrypt
Please enter your encrypted string here:
Please enter a secret key:
DECRYPT
Decrypted Message

The Decrypt page is where the user would enter their encrypted string and their secret key to decrypt and receive the secret message.

Scan Page:



The scan page is where the user can choose and upload two images and compare their MD5 hash values. This may help identify that something has changed in the image if the original is used in the comparison.

Project Review

Amount Achieved

Outlined in the research document previously, the core functionality of Stegano is below:

1. Embed a secret message.
2. Extracting a secret message.
3. Encrypting a secret message prior to embedding.
4. Decrypting a secret message after extracting.
5. The ability to compare hash values of image files.

The above core functions were achieved, and all functions work well.

Extra features include the following:

- A customised app icon on android
- Help pages for the core functions.
- An app which works on android mobile devices and not just on the emulator software.
- A fully functional production frontend which is ready for the app store.

Not Achieved

Originally, the app was going to use benford's law to try and detect steganography. Benford's law can detect the probabilities of highly likely or highly unlikely frequencies of numbers in datasets. The algorithm would have been able to detect if a file is a suspicious stego-carrier by analysing the bits in each byte to. Using a generalized form of benfords law, we should have been able to perform a statistical attack on an image to tell whether or not it is a stego carrier. However, due to time constraints and a difficult year it was too much of a difficult challenge to undertake. Going forward, I would work to implement this feature in the app since it is unique and no other steganography app has this ability.

Problems Encountered

Creating an app in Android Studio proved to be a challenge at times. This was mainly due to the fact that certain types of errors don't show up when you compile and run the emulator, but these cause the app to crash when it runs.

Throughout the production of Stegano there were many problems and unforeseen challenges. My first task when starting to make the app was to tackle the basic embed and extract function of the app. Since the approach I choose to go with was the LSB approach, it was a challenge to properly implement the functions. I decided to code these functions in Eclipse first and then migrate the code over to Android Studio. However, I noticed quickly that to make the code work in Android studio it would require extra functions and code just to make it compatible with the most current android version.

Another problem encountered was during the creation of the embed and extract function. More specifically when trying to save and image to the gallery. During the early days of the project, the button to save an image kept crashing the app when a user would try to save an image. After investigating and debugging the issue, it turned out to be related to a missing piece of code in the Android Manifest File. Even though I granted the app permissions to read/write on the phone, I needed to explicitly state those permissions.

Problems arrived when trying to create the embed function of the app. I knew I needed to embed the secret message using the LSB approach, however, I did not know that I needed to compress the secret message as the first step. After hours of research and after reading the following research paper: steganography , I was able to identify my error.

When creating the MD5 hash checker the MD5 hash failed to compute the correct value for the second image. It used to produce the hash value of the first image again. After investigating the issue, it turned out that I needed to add if statements in the md5 function to make it identify which filepath it was meant to use. Before this it appeared to not update the filepath when the user chose the second image. Since the filepath was the same as the first image, it produced the same result. After adding a 3 if statements into the function, it worked as intended.

The frontend of the app suffered problems during the development phase too. Mainly due to my inexperience with android studio and failing to understanding the different layouts that can be used. When adding buttons and textview areas, they would appear to be shifted far to the left when the app was run even though it showed that it was on screen in the design tab. This turned out to be related to the way I added the constraints to each button and textview. After changing the layout to a relative layout, I was able to just drag and drop buttons and textviews and they would display correctly on screen.

Testing

This app was tested upon completion of all functions. For example the embed/extract function was tested when a user tried to misuse the app and try to upload something other than an image. The app only allows file which end in image extensions (jpg/png) and that are in the photo section of the gallery. It does not allow the use of any others and produces a correct error message. There is no chance that a user can navigate to another folder to attempt to upload something other than an image.

Input fields were also tested, for example how the app handles blank input fields when trying to emb/extract/encrypt or decrypt. The user must enter text into all fields and is warned/notified that text is needed. The app will not process blank textfields and therefore the user cannot attempt to break the app this way because before attempting to compute any of the functions it first checks the textfields for strings, if it doesn't find any then it returns and notifies the user.

Since the only user input is textfields and image uploads, there was no other area to break the app.

All functions of the app work as they should.

Difference from Design

As mentioned before, benford's law could not be implemented. Also compared to the design doc, Android Studio did no support such a frontend like depicted there using adobe photoshop, however, I find the current frontend to be suitable for the task at hand.

What I learned

Throughout this project I learnt a lot about the various steps involved when software developing. I have learnt a lot about how java and how android studio implements it and Iam happy to say that my skills as a programmer and as a android developer have improved. The overall insight into the java development from frontend to backend has been a big learning experience due to the fact that it comprised of most of my project if not all.

If I had to redo the work and start over I would create a plan which includes time for fixing issues. My plan did not include time for me to fix broken code, which led me to miss out on key deadlines on specific dates. Out of all the problems I faced, time management was certainly a big one. A note to make also is that if I had the ability to start over and redo my work, I would do so not in the middle of a Pandemic 😊.

Conclusion

In conclusion, I am happy with the overall progress made for my final year project. The core functionality of Stegano works as it should and it works on android phones too. Although the app certainly doesn't look professional, it gets the job done. Benford's law is something that wasn't done; however, this is something that will be looked at in the future as I feel it is an important aspect which makes this app unique. Overall I am happy with my experience and the amount I have learned from this project.