Polytechnic University of Puerto Rico
Electrical & Computer Engineering & Computer Science Department
Hato Rey, PR

Software Requirements Specification

Lambda Solutions Group Presents:

StegSleuth

Name	Student ID	Program	Signature
Emilio Acosta Ortiz	86144	Computer Engineering	
Kevin Medina Santiago	85540	Computer Engineering	
Gabrielys Rivera Flores	89456	Computer Engineering	
Manuel Seda Batista	120758	Computer Engineering	
Ricardo Vilá Palacios	113924	Computer Engineering	

Changelog

Name	Date	Changes Performed	Version
Manuel Seda Batista	March 25th, 2019	Created a LaTeX template	0.1
Gabrielys Rivera Flores	April 6th, 2019	Started sections 2.2 & 2.3	0.1.1
Emilio Acosta Ortiz	April 7th, 2019	Started section 3.1	0.1.2
Ricardo Vilá Palacios	April 7th, 2019	Started sections 3.4 & 3.5	0.1.3
Kevin Medina Santiago	April 15th, 2019	Started sections 1 & 2.1	0.1.4
Manuel Seda Batista	April 20th, 2019	Started sections 2.4, 2.5 & 2.6	0.1.5
Kevin Medina Santiago	May 5th, 2019	Combined parts into rough draft	0.1.6
Manuel Seda Batista	May 10th, 2019	Moved draft into LaTeX template	0.2
Gabrielys Rivera Flores	May 14th, 2019	Made corrections to sections 2.2 & 2.3	0.2.1
Emilio Acosta Ortiz	May 19th	Made corrections to section 3.1	0.2.3

Contents

I. List of Figures

II. List of Tables

1. Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

1.1. Purpose

This document is purposed with describing the requirements for the *Steg-sleuth* software that will be developed. A detailed description of system constraints, interface, and interactions with other external applications will be exposed. This writeup is done primarily for the customers convenience and for the them to approve or change what will be developed.

1.2. Scope

The *Steg-sleuth* (final name pending) software is a desktop application with the express purpose of assistance of solving steganography related puzzles and challenges. The application will run primarily on UNIX systems as it will require access to Terminal commands on the back-end. It will be open source and free to download primarily by Polytechnic University of Puerto Rico alumni as well as anyone else. Users will be able to analyze files for hidden data such as text or other documents as well as the ability to hide data in files themselves. This will be done through steganographic algorithms such as least significant bit (LSB). Since this data may be encrypted then cryptographic methods will also be included for decryption and encryption of documents and text. File can be manually analyzed through either a text editor for its ASCII data or through a hex-dump for raw binary data. Finally, in order to verify file integrity, users will be able to compare files through their hashes in the application.

1.3. Definitions, Acronyms and Abbreviations

Term	**Definitions**
Binary	Positional numeral system with a base of 2
Carrier	The file that will hold the hidden file
Encryption	The process of obscuring information to make it undecipherable
Hash or Message Digest	A value created from a file passing through a hashing formula. Used to verify file integrity
Hexadecimal	Positional numeral system with a base of 16
Hex-dump	Hexadecimal view of file data
Payload	The file to be hidden
POSIX Compliant	A software that is developed following a family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems
Steganography	The practice of concealing a file or message within another file. Usually done by changing certain bits in the carrier file to match those of the payload
UNIX	A multi-user, multitasking computer operating system

Table 1.1: Terms & Definitions

Term	Acronym Definitions
ASCII	American Standard Code for Information Interchange
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
LSB	Least Significant Bit
OS	Operating System
POSIX	Portable Operating System Interface
PUPR	Polytechnic University of Puerto Rico

Table 1.2: Terms & Acronym Definitions

1.4. References

The following references use the IEEE citation format.

[1] Margaret Rouse, Allan Leake and Adam Hughes, "Definition Database." https://searchsqlserver.techtarget.com/definition/database. Online, Acessed: 2018-12-15.

[2] Dictionary.com, "etcetera." https://www.dictionary.com/browse/etcetera. Online, Acessed: 2018-12-15.

[3] Margaret Rouse, "Definition Hardware." https://searchnetworking.techtarget.com/definition/hardware. Online, Acessed: 2018-12-15.

[4] TechTerms, "Hash Definition." https://techterms.com/definition/hash. Online, Acessed: 2018-12-15.

[5] w3schools, "Introduction to HTML." https://www.w3schools.com/HTML/html_intro.asp. Online, Acessed: 2018-12-15.

[6] IEEE, "What is IEEE?." https://supportcenter.ieee.org/app/answers/detail/a_id/190/~/what-is-ieee%3F. Online, Acessed: 2018-12-15.

[7] Oxford Dictionary, "Definition of interface in English." https://en.oxforddictionaries.com/definition/interface. Online, Acessed: 2018-12-15.

[8] Wikipedia contributors, "Puerto Rico Maritime Transport Authority." https://en.wikipedia.org/wiki/Puerto_Rico_Maritime_Transport_Authority. Online, Acessed: 2018-12-15.

[9] Margaret Rouse, "Definition QR code (Quick Response code)." https://whatis.techtarget.com/definition/QR-code-quick-response-code. Online, Acessed: 2018-12-15.

[10] Margaret Rouse, "Definition Query." https://searchsqlserver.techtarget.com/definition/query. Online, Acessed: 2018-12-15.

[11] TechTerms, "Server Definition." https://techterms.com/definition/server. Online, Acessed: 2018-12-15.

[12] Margaret Rouse, "Definition Software." https://searchmicroservices.techtarget.com/definition/software. Online, Acessed: 2018-12-15.

[13] techopedia, "Uniform Resource Locator (URL)." https://www.techopedia.com/definition/1352/uniform-resource-locator-url. Online, Acessed: 2018-12-15.

[14] Collins English Dictionary, "Definition of 'web page'." https://www.collinsdictionary.com/dictionary/english/web-page. Online, Acessed: 2018-12-15.

[15] Inflectra, "What are System Requirements Specifications/Software (SRS)?." https://www.inflectra.com/ideas/topic/requirements-definition.aspx. Online, Acessed: 2018-12-15.

1.5. Overview

The remainder of this document includes 3 chapters. The second chapter is a definition of system functionality, constraints and assumptions for the final product. The third chapter focuses on the requirements for the application.

2. Overall Description

This section will give an overview of the application. The application will be dissected and explained on how it functions and how it interacts with other systems. At the end, constraints and assumptions will be listed and accounted.

2.1. Product Perspective

The product will be a desktop application with a front end GUI. The primary function of the system is to allow the user to select a file as the carrier for a steganographic analysis be it for extraction or insertion of a payload. Many different types of algorithms exist for this process, with most of the available ones involving changing the least significant bits of the carrier to match those of the payload. The application will allow the user to select from various algorithms to achieve this. As an extra measure of confidentiality people sometimes encrypt the payload before storing it in the carrier. If encrypted hidden information is stored in the carrier but is not decrypted then what will be extracted will be an indecipherable mess. For this reason, the application will have the option to either encrypt the payload before being hidden in the carrier or decrypting it after being extracted. Since a large amount of cryptographic algorithms can be used for this, the application will allow the user to choose one from a selection and to insert a key with which to encrypt or decrypt. In the event that the user would like to verify the integrity of a supposed carrier file against a similar file the application would help in that regard. In order to achieve this comparisons are done through hashing values. The user will select two files of the same extension and select a hashing algorithm from the various options which will get the message digest for both items. If both files have the same value then they are truly equal. If both values are different then they are different. Sometimes a payload could be something as simple as some text hidden in the data of the carrier. In order to verify this, two methods of viewing a text output will be done. One method is by showing the carriers text output in a regular text box. After receiving the text output the user will be able to either scroll through the data or do a string search for something specific. The other method of verifying the data is through a hex-dump. Through the hex-dump the user will be able to view both the hexadecimal data of the file and the ASCII values as well.

2.1.1. System Interfaces

2.1.2. User Interfaces

2.1.3. Hardware Interfaces

2.1.4. Software Interfaces

2.1.5. Communication Interfaces

2.1.6. Memory Constraints

2.1.7. Operations

2.1.8. Site Adaptation Requirements

2.2. Product Functions

The main purpose of the StegSleuth tool is to assist students of the Polytechnic University of Puerto Rico who participate in Capture the Flag (CTF) competitions. StegSleuth will be a tool for steganography that will offer the following functions:

- **Find hidden payloads in a carrier** - The payload is extracted from the carrier using the algorithms provided.

- **Hide payloads in a carrier** - The payload is embedded in the carrier using the algorithm provided.
- **Encrypt payloads to hide in a carrier** - Encrypt payloads that are going to be embedded in a carrier.
- **Decrypt hidden payloads in a carrier** - Decrypt payloads that are extracted from a carrier.
- **Compare carriers to see if there are changes** - Compare carriers to verify if there is any hidden information.

In figure 2.1, the relationship between the system requirements can be observed. Also, figure 2.2 shows the use case diagram that models the functionalities of the system in different use cases.

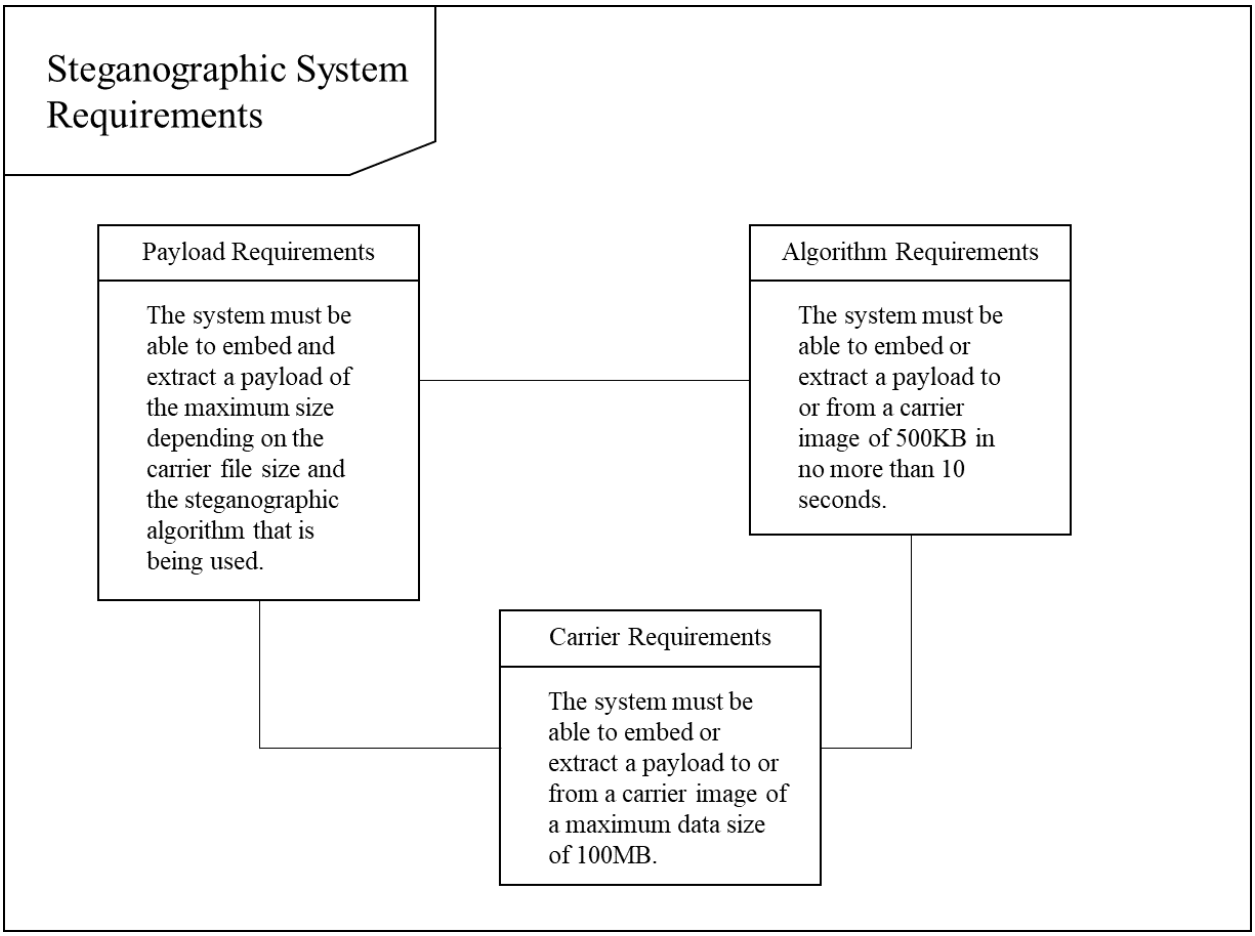

Figure 2.1: Requirements Diagram

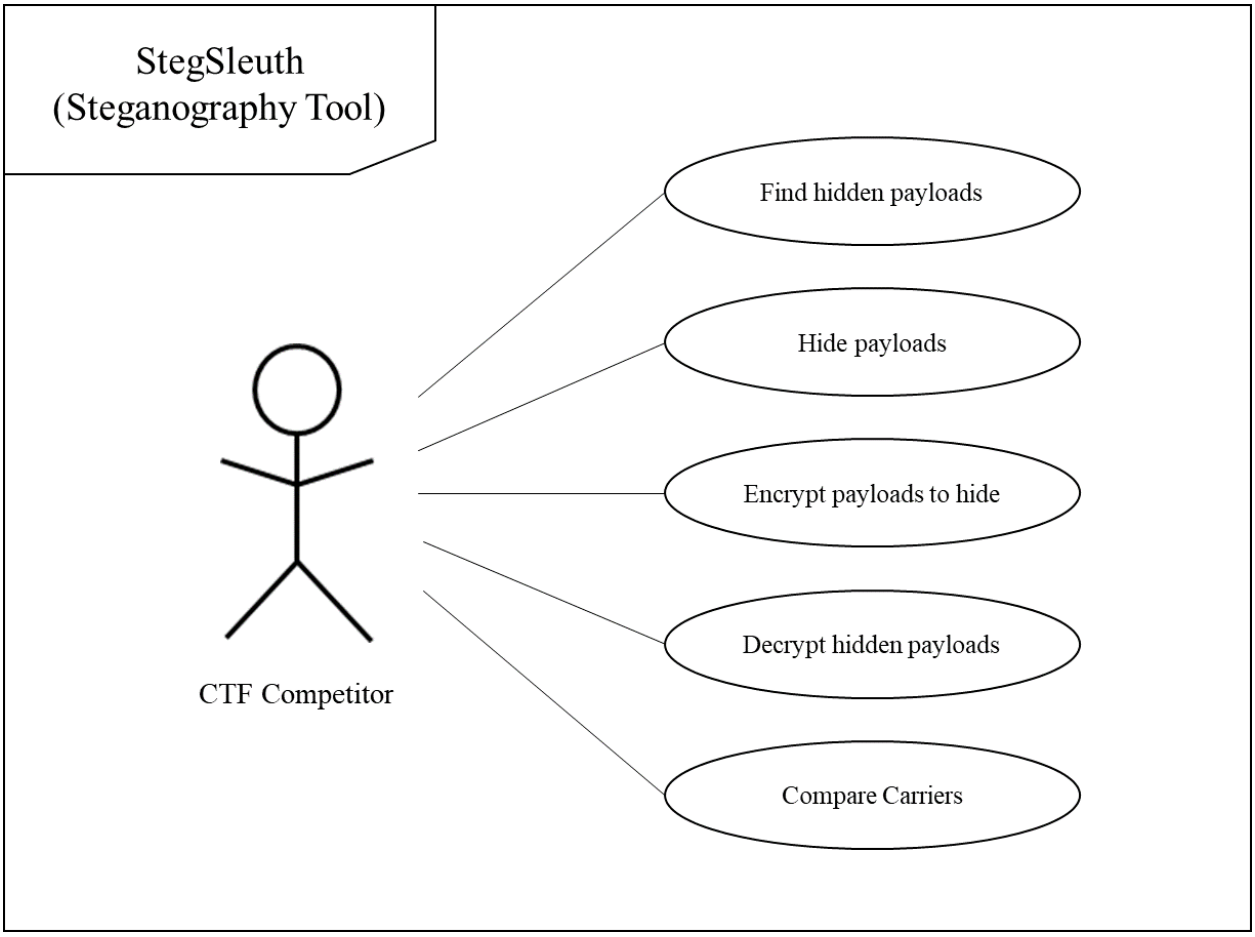

Figure 2.2: Use-case Diagram

2.3. User Characteristics

This tool's target demographic will be cyber-security Capture the Flag competitors. The users should be able to utilize all of the functions of the system at their convenience. Said users should have knowledge in the field of steganography and cryptography and may be of any age or academic preparation. With this preparation they should be able to decide which cryptographic and steganographic algorithms they will be using to solve their problem. In addition, they must know how to operate the Linux operating system (OS).

2.4. Constraints

The application is mostly limited by the file types that the user will choose. Depending on the progress being made on the project once image steganography is resolved in its entirety then other file types will be added progressively. Image steganography algorithms are highly available and simple to implement, so that is what it will start by. Since the application will be developed with a mindset that the users primary goal is to participate in Capture the Flag competitions, this application is being developed for an audience who uses Linux distributions as their Operating System of choice for these events. As such, the application is being designed with the Python programming language for the Linux line of operating systems.
Terminal commands are a big part of this application, as functions such as the string searches and the cryptographic algorithms run through it. As such, making sure that these functions are installed in the users computer is important. In the event that the necessary commands are not present, a stable internet

connection and user authorization must be present in order to install them.

2.5. Assumptions and Dependencies

The following assumptions and dependencies shaped the way the project was conceived. Any major change will cause a redesign of the system.

- **Python 3 programming language** - StegSleuth will be developed using Python 3, it is assumed that the user's machine will be compatible with said operating system; Python 2 is fundamentally incompatible with the above version and thus will not be used or supported
- **Linux-based operating system** - although StegSleuth will be developed with POSIX compliance in mind, the software will only be tested and validated on Linux-based systems; other POSIX compliant operating systems may be able to run the software, but they are not guaranteed to run the software without problems, as testing these scenarios are not within the scope of the project
- **Python steganography libraries** - the purpose of this project is to create a front-end for steganography tools that will be used for PUPR's cyber-security competitions, therefore, pre-existing Python steganography libraries will be used for the bulk of the operations to be carried out by the software; these libraries are considered a hard dependency for the project
- **input correctness** - because of the complexity of steganography algorithms, no attempts will be made to correct or interpret invalid user input, therefore, valid input formats given by the user are assumed to contain steganographic data and will be processed; if the user input cannot be processed, it will be rejected or the user will be given an error message
- **computer specifications** - the user is assumed to be using a relatively recent machine to run the software (less than 5 years old), with an x86-64 processor; no attempts will be made to support other processor architectures or older machines within the execution of the project and no testing will be done on older machines

2.6. Apportioning of Requirements

The following requirements are seen as nice-to-have and will be placed here as future requirements.

- **macOS compatibility** - because of macOS's POSIX compliance, compatibility with StegSleuth should be achievable with minimum or no changes to the software's code-base; because of this, testing is the minimum effort expected for this task
- **Windows port** - although compatibility layers such as Cygwin exist to provide a POSIX compliant interface within Windows, porting to this software platform is expected to require at least some modification to the code-base; at worst, a new software branch will be needed to acomodate this operating system

3. Specific Requirements

3.1. External Interfaces

Name of item	**New Project Options**
Description of Purpose	Allows the user to search the image to be worked on
Source of Input or Destination of output	The user selects through a button called "browse" the image that he wishes to work
Timing	The waiting time of the application should not be more than 5 seconds approximately.
Command Format	The New project option only responds when the user selects the image they want to use and presses the confirmation button.

Table 3.1: New Project

Name of item	**File Explorer**
Description of purpose	Allows the user to load the image that they want to work with, it can be loaded from the computer's internal storage or an external storage device
Input source or output destination	The user can place the file-path of the image in a text box provided in order to find the image in a more accurate way
Valid range, accuracy and/or tolerance	Text box: can not exceed 30 characters
Relationship with other inputs/outputs	This interface depends on the user having selected an image either from his computer or from an external storage device
Data format	The text must be encoded in ASCII
Command format	The interface will only respond once the text field is completed or the image is selected; after this, the user must press the load image button
Final messages	Once selected, the image will appear with the message: "Image Loaded" and will then transfer it to the next GUI

Table 3.2: Loading an Image

Name of item	**Save As Interface**
Description of the purpose	It allows the user to save the result obtained in the computer or in a USB
Input source or output destination	The user can name the file extracted from the image in a text box provided
Valid range, accuracy and/or tolerance	Text box: cannot exceed 30 characters
Relationship with other inputs/outputs	This interface depends on the user having selected the destination where he wants the information extracted from the image to be stored
Data format	The text must be of alphanumeric values
Command format	The interface will only respond once the text field is completed. After this, the user must press the button that will say "save"
Final messages	Once placed the place where you want the file to be saved and once the name is placed there the user must press the button that will say save once the button is pressed, a message will be sent saying "File saved"

Table 3.3: Saving an Image

Name of item	**Encode/Decode Interface**
Description of purpose	It allows the user to select between the Encode and Decode operations through a few button radios
Input source or output destination	The user will be able to select these operations through button radios, so that he can continue to the next phase
Valid range, accuracy and/or tolerance	Radio button: will consist with a predetermined function; common button: will consist with a default function
Relationship with other inputs/outputs	This interface depends on the user having correctly selected the operation he wants to perform (Encode, Decode) and then pressing the next button
Data format	Predetermined
Command format	The interface will respond when the user has correctly selected the operation you want to perform (Encode, Decode) and then press the next button

Table 3.4: Encode/Decode Interface

Name of item	Text/Algorithm Interface
Description of purpose	Allows the user to select between text, algorithm and steganography operations
Input source or output destination	The user can select these operations through a check box button to then enable other system functions
Valid range, accuracy and / or tolerance	Text box: cannot exceed 30 characters; Combo box: allows the user to select one of the operations already predetermined in it; common button: will consist with a default function; check box button: will enable the function associated with it
Relationship with other inputs/outputs	This interface depends on the user having correctly selected the operation he wants to perform and then pressing the next button
Data format	The text must be of alphanumeric values
Command format	The interface will respond when the user has correctly selected the operation they wish to perform and then press the next button

Table 3.5: Text/Algorithm Interface

Name of item	Text Editor Interface
Description of purpose	Allows the user to place or view strings in the image being worked on, depending on the mode of operation: encoding or decoding, respectively
Input source or output destination	The user's keyboard or the image itself is the input (depending on encode/decode); The strings shown on the textbox is the output
Valid range, accuracy and/or tolerance	Text box: no more than 30 chars; common button: predetermined function
Relationship with other inputs / outputs	This interface depends on the user having correctly selected the operation he wants to perform and then pressing the next button
Data format	Alphanumeric values
Command format	The interface will respond when the user has correctly selected the operation they wish to perform and then press the next button

Table 3.6: Text Editor Interface

Name of item	Find String Interface
Description of purpose	Allows the user to search for a specific string sequence
Input source or output destination	The user's keyboard; The strings shown on the textbox is the output
Valid range, accuracy and/or tolerance	Text box: no more than 30 chars; common button: predetermined function
Relationship with other inputs / outputs	This interface depends on the user having correctly selected the operation he wants to perform and then pressing the next button
Data format	Alphanumeric values
Command format	The interface will respond when the user has correctly selected the operation they wish to perform and then press the next button

Table 3.7: Find String Interface

Name of item	Close Program Interface
Description of Purpose	Allows user to close the application
Timing	The waiting time of the application should not be more than 5 seconds
Screen Format	40% of the screen will be used to display the item list; the rest will have buttons, drop-down lists, and text fields for other interfaces

Table 3.8: Close Program Interface

3.2. Functions

3.3. Performance Requirements

The following are the performance requirements that have been set for StegSleuth's functionality. Because of StegSleuth's nature as a toolkit, these requirements will be adjusted throughout the project's development.

- **GUI responsiveness**: The GUI must respond quickly to user input. Any delays must be imposed by complex operations such as decryption or steganographic encoding and not because of GUI hangs or other GUI related sources of delays.
- **Prompt execution of steganography operations**: Steganographic operations must be executed as close as possible to the time it takes to complete the algorithm's work. Overhead introduced by StegSleuth's operation should be kept to a minimum.
- **Decryption speed**: Encrypting and decrypting are complex proceedures whose execution times depend on the encryption algorithm used and the manipulated file's size. Since the files used in the intended real-world applications for StegSleuth are small, the main sources of delay are expected to come from the encryption algorithms themselves. StegSleuth must once again provide as little overhead as possible.
- **Integration with file-system**: Operations within the file-system are expected to be quick and efficient. This aspect is to be taken care of via Python file-system libraries and system calls when needed.
- **General program speed**: StegSleuth is designed with efficiency in mind. Therefore, unnecessary complexities are to be avoided to provide a fast program.

3.4. Design Constraints

Minimum System Requirements (Assuming Ubuntu 16.04)

- 2 GHz dual core processor
- 2 GiB RAM (System memory)
- 25 GB System Hard drive space
- VGA capable of 1024x768
- Either a cd/drive or USB port for the installer media
- Display
- 2 GB of V-Ram for physical installs (or 4 GB for virtualized installs)
- 3D acceleration capable video card with at least 256 MB
  - Visual Effects
  - Intel – i915 or better
  - Nvidia (with proprietary driver)
  - ATI (with proprietary driver)

3.4.1. Standards Compliance

- Encryption standards must be better or equal to AES256, RSA and blowfish
- Developed and documented in English
- Objects analyzed will be images with the following file extensions: jpg and png

3.5. Software System Attributes

3.5.1. Software Dependencies

- Requires python 3.5+ and pip version 9.0 or higher
- For GUI requirements refer to Minimum system requirements

3.5.2. Reliability

- The system shall not crash due to any external input to the application, much less from any internal application error. If a system error should occur a graceful error message will be displayed, giving the user some guidance as to what is failing.
- Time benchmarks should meet at least 10 seconds of computation time for analyzing a 175kb image in a 500kb image

3.5.3. Security & Privacy

- The system won't be vulnerable to the same inputs it is trying to analyze. So it will be assumed that the payloads that are being analyzed will not affect the underlying OS in any way.
- Application must not be able to escalate its own user privileges

3.5.4. Training-related Requirements

Users should be familiar with steganography and cryptography

3.5.5. Maintainability

The code should be fully documented in order to correct any future issue with the application.

3.6. Modularity

The application will feature modular code so as to add new analyzing techniques easily and quickly.

3.6.1. Portability

The application will be developed using a Qt base python package, making the app a Cross-Platform application capable of running in Microsoft Windows, Linux and MacOS.

- MacOS - versions depend on package chosen
- Linux - are mobile and/or lightweight GUI versions supported (would be nice to be able to run in a small chip installable anywhere ex. Email servers)
- Windows - versions depend on package chosen

3.6.2. Packaging Requirements

The source code will be packaged along with all code documentation. Also a README file will be included for basic instructions on how to run the application and use it.

3.6.3. Legal Requirements

The app recognizes third party packages used and complies with all their licenses