SOFTWARE REQUIREMENTS SPECIFICATION

for

A Secure method to send Emails using - Compression, Encryption and Pixel Value Differencing

Version 1.0 not approved

Prepared by
CSU 152 24 MDL15CS047 Joby Mathew
CSU 152 34 MDL15CS070 Merin Francis
CSU 152 39 MDL15CS087 Ria Rajan Alappat
CSU 152 56 MDL15CS113 Tony Josi

November 6, 2018

Contents

1 Introduction

1.1 Purpose

Electronic mail (email or e-mail) is a method of exchanging messages (”mail”) between people using electronic devices. Email security refers to the collective measures used to secure the access and content of an email account or service. It allows an individual or organization to protect the overall access to one or more email addresses/accounts. An email service provider implements email security to secure subscriber email accounts and data from hackers - at rest and in transit.

This project aims at creating a platform for securing the sending of mails using Compression, Encryption and Pixel Value Differencing

1.2 Intended Audience and Reading Suggestions

The document is intended for developers, project managers, testers and documentation writers. The rest of this SRS contains further details of the project, which includes scope of the project, as well as software and hardware requirements.

1.3 Project Scope

The product is useful for securing the existing mailing system by using most efficient data security methods like Compression(by arithmetic coding), Encryption and Pixel Value Differencing. The arithmetic coding helps to compress the data size as well thus reducing the bandwidth and embedding requirements.

1.4 Overview of Developer’s Responsibilities

The responsibility of the developer is to implement the further features :

- Implement arithmetic coding to compress the given mail content
- Develop 128 bit AES to encrypt the encoded data after compression
- Implement a stenographic algorithm to embed encrypted data using LSB substitution and Pixel Value Differencing
- To implement a mailing system to send mails using the proposed security method
- Develop a system to decode the cipher text from image
- Impement inverse AES to get ecoded textual data
- Reverse compression to obtain the orginal email text

2 Overall Description

2.1 Product Perspective

The proposed project is a modification of the original mailing platform used for sending emails by improving the security and efficiency. Data compression involved in this project compresses the total data that needs to be send across and thus increases the efficiency of further processes. AES and Steganography further adds the security of the current system.

2.2 Product Functions

- Input the details of the mail including content and sender
- Apply the proposed methods to make the data secure
- Send mails with the processed data to the given sender address

2.3 User Classes and Characteristics

The application is intended to be used by regular users, Who may not possess any technical expertise and it aims at improving Quality Of Experience of users who are in need of sending emails more securely and efficiently. The user simply need to have the basic knowledge to give information to the respective field.

2.4 Operating Environment

The product can be accessed by terminal commands on either Mac, Linux or Windows

- On any Operating system containing the required libraries there is a requirement of functioning internet connection.

2.5 Design and Implementation Constraints

The Design and Implementation Constraints include:

- There will be a limit on the size of the data which can be embed to the cover image depending upon the cover image.
- Constraint on the resource utilization while using AES algorithm
- Only textual data can be given as input to the system

2.6 User Documentation

This system doesn't require any user documentation since the user can simply type in the command and input data accordingly as prompted

2.7 Assumptions and Dependencies

- The designed algorithm is for English language and the text the application parses must be written in this language.
- The interface of resulting system will be easy to use and accessible without a time or location constraint.
- The application requires a stable network connection.
- The platform should be containing the updated python versions of cv2, docopt and numpy.

3 Use Case Diagram

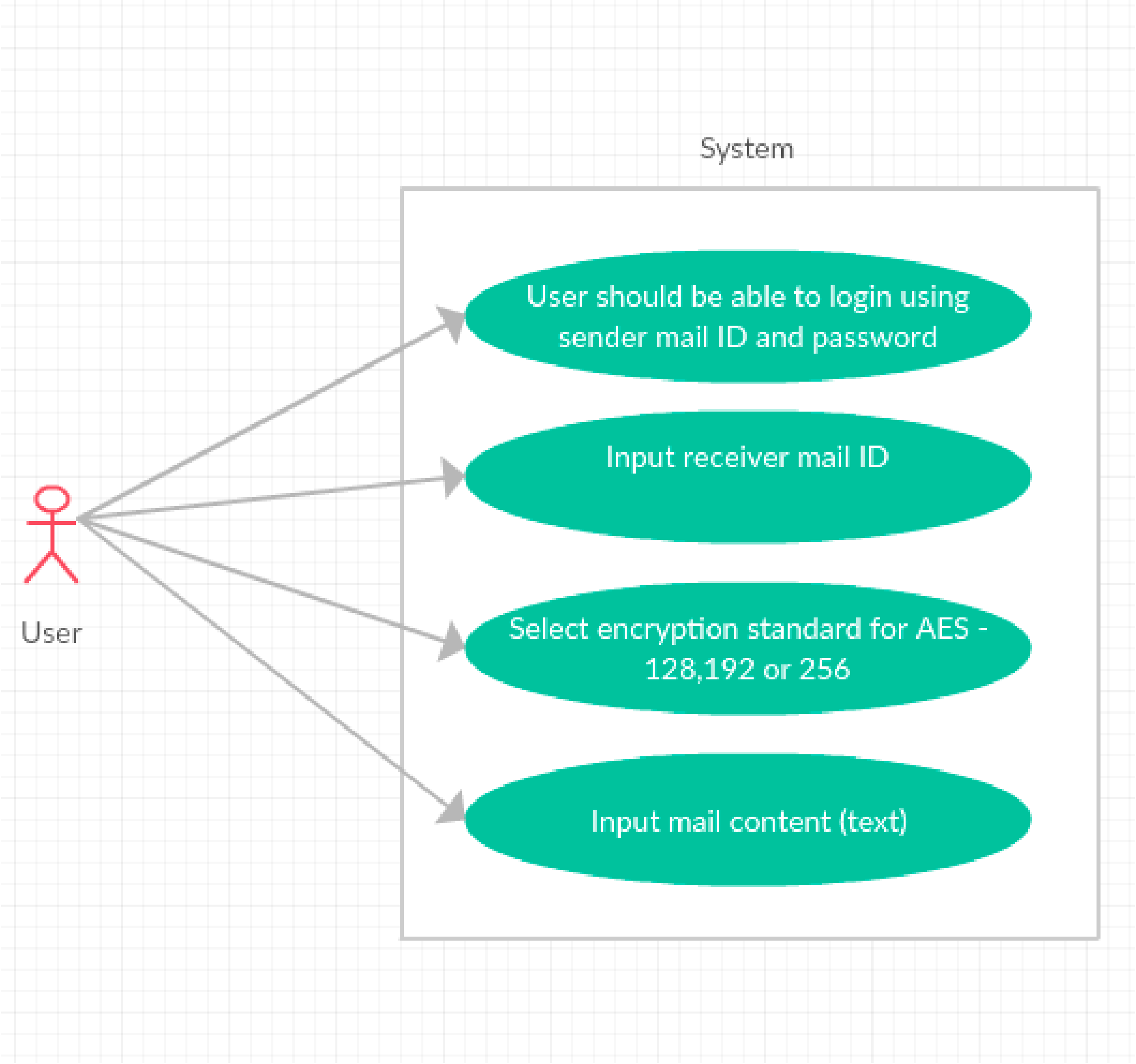
System
User should be able to login using sender mail ID and password
Input receiver mail ID
Select encryption standard for AES - 128,192 or 256
Input mail content (text)
User

Figure 3.1: Use Case Diagram

4 External Interface Requirements

4.1 User Interfaces

The user interface is a simple command that further prompts user about the further data that is to be taken as input. This input data will be processed and then send to the mail of the given sender. The mail retrieval also takes place in the same way by using same command.

4.2 Hardware Interfaces

No particular hardware interface.

4.3 Software Interfaces

This application takes input from the user as textual data of sender address (mail) and content of the mail (text). The input is then sent for further processing of text compression, encryption and stenography with a given cover image. The processed data is then sent across the network to the given address. The application also makes use of APIs like cv2, docopt, numpy, etc....

4.4 Communications Interfaces

The only communication over the network is for sending the mail with the processed data. The mailing system uses Simple Mail Transfer Protocol (SMTP) for sending mails.

5 Hardware and Software Requirements

5.1 Hardware Requirements

- System: Any Quad core system clocked at 2.4GHz (Minimum)

  AES and Stenography process consumes a lot of system resources, Therefore a system with Clock speed at least 2.4GHz is required.

- RAM : 4GB (Recommended)

  For smooth and faster functioning of AES.

5.2 Software Requirements

- Operating system : Linux (64-bit)

  Developer friendly,Powerful shell,Support,Flexibility.

- Language : Python 3.0

  A large number of APIs for various purposes is available in python.

- Libraries : cv2, docopt, numpy

  cv2 - Python library for image processing.

  docopt - Command-line interface description language

  numpy - fundamental package for scientific computing with Python.

6 Functional Requirements

6.1 Mailing System Requirements

- The system should be able to send textual mails to the given addresses after performing the required functions
- The input should only accept textual data
- The system should not preserve the text formatting

6.2 Input Requirements

- The system should be able to define a maximum limit for the size of input data
- The input should only accept textual data
- The system should not preserve the text formatting

6.3 Compression Stage Requirements

- The system should be able compress the given text file using arithmetic coding
- The compression algorithm must accept the acceptable input size prescribed for the input
- The output file should also be a text file

6.4 Encryption Requirements

- The algorithm used should be AES
- The cipher kep shoud be 128 bits with block size of 128 bits

6.5 Stenography

- The system should be using Pixel Value Differencing algorithm for embedding encrypted data to the cover image
- The cover image should be colour image
- The output should be also an embedded colour image

7 Other Nonfunctional Requirements

7.1 Performance Requirements

Calculation time and response time should be as little as possible, because one of the software's features is timesaving.The capacity of servers should be as high as possible.

7.2 Safety Requirements

The possible harm that can occur during the process is the chance of detection of data from cover image using steganalysis, but the chances are minimal due to the use of Pixel Value Differencing algorithm.

7.3 Security Requirements

The data given by the user is undergone several steps for improving security before transmitting across the network.

Bibliography

[1] A. Cheddad, et al., ”Digital image steganography: Survey and analysis of current methods,” Signal processing

[2] S. Atawneh, et al., ”Steganography in digital images: Common approaches and tools,”