Software Requirements Specification (SRS)

# Project Title: Secure Image Transmission Using Cryptography

Author: Abhishek Kumar Gupta  
Date: July 2025

# 1. Introduction

## 1.1 Purpose

This software aims to ensure secure transmission of images by using cryptographic algorithms like AES and RSA. This application is useful in areas such as military communication, secure photo sharing, or medical image transmission.

## 1.2 Scope

Encrypt and decrypt image files securely. User authentication for image access. GUI-based application (Python Tkinter). Export encrypted images for sharing.

## 1.3 Definitions

- Cryptography: The practice of secure communication in the presence of adversaries.  
- AES: Symmetric key encryption algorithm.  
- RSA: Asymmetric public-private key encryption.

# 2. Overall Description

## 2.1 Product Perspective

Standalone Python-based desktop application.

## 2.2 Product Functions

- Upload image  
- Encrypt image  
- Save encrypted image  
- Decrypt image with key  
- GUI supports drag/drop

## 2.3 User Characteristics

Basic computer skills. No cryptography knowledge required.

## 2.4 Constraints

- Desktop Python app (not mobile/web).  
- Max image size 10MB for performance.

# 3. Specific Requirements

## 3.1 Functional Requirements

- FR1: User uploads image.  
- FR2: User selects encryption method (AES or RSA).  
- FR3: Encrypted image is saved.  
- FR4: Decryption key required to decode image.  
- FR5: GUI supports drag/drop.

## 3.2 Non-Functional Requirements

- NFR1: App must work on Windows/Linux/Mac.  
- NFR2: Encryption must be completed within 5 seconds.  
- NFR3: Application must handle at least 100 images in a session.

# 4. External Interface Requirements

## 4.1 User Interfaces

Built with Tkinter: File dialog, buttons, image preview.

## 4.2 Hardware Interfaces

RAM ≥ 4GB  
OS: Windows/Linux/Mac

## 4.3 Software Interfaces

Python 3.x  
Libraries: tkinter, cryptography, PIL, base64

# 5. Appendices

- Sample test image  
- Key management flow  
- Encryption logic (AES block 128-bit)