**Vinaayak G Dasika, PES2UG24CS588**

**Yakshith Naidu, PES2UG24AM186**

**Y. Sai Aneesh Reddy, PES2UG24CS615**

**Yashita Anand, PES2UG24CS613**

**File Encryption using NCURSES, TEAM 16**

# Software Requirement Specification (SRS) Document

## FileCryption: File Encryption/Decryption Tool

### 1. Brief Description of Project

FileCryption is a console-based application developed in C that provides secure file encryption and decryption capabilities. It features an intuitive text-based interface allowing users to browse their file system, select files for encryption or decryption, and protect their data with password-based encryption using strong cryptographic algorithms.

### 2. Purpose / Goal

The primary goal of FileCryption is to provide users with a simple yet secure method to protect sensitive files through encryption. The system aims to make file security accessible without requiring deep cryptographic knowledge, while still implementing robust security measures using industry-standard encryption algorithms.

### 3. Usefulness / Benefit

- **Data Security:** Protects sensitive files from unauthorized access through strong encryption.
- **Ease of Use:** Provides intuitive file browsing and simple password-based encryption/decryption.
- **Minimal Resource Usage:** Operates efficiently with minimal system resources.
- **File Type Recognition:** Recognizes and displays file types to help users identify files.

## 4. Hardware / Software Involved

**Hardware Requirements:**

- Standard PC/Laptop
- Minimum 256MB RAM
- 50MB disk space

**Software Requirements:**

- Programming Language: C
- Operating System: Unix/Linux/macOS
- Libraries:
  - libsodium (for cryptography)
  - NCURSES (for text user interface)
  - libmagic (for file type detection, included on Unix/Apple systems)
- Compiler: GCC or compatible C compiler

## 5. Detailed Feature List

**File Management:**

- File system browsing with directory tree view
- Hidden file display toggle
- File type recognition and display

**Encryption:**

- Password-based file encryption
- Secure key derivation from passwords
- Authenticated encryption using XChaCha20-Poly1305

**Decryption:**

- Password-based file decryption
- Automatic detection of encrypted files (.enc extension)
- Verification of file integrity during decryption

**User Interface:**

- Text-based user interface with menu system
- Interactive file browser
- Keyboard navigation (arrow keys, enter, escape)
- Status messages and confirmations

**6. Test / Demonstration Plan**

- The application can be tested by creating some test files and testing if the encryption and decryption is successful.

**7. Expected Interaction Interface and Sample Use Cases**

**Interaction Interface:**

- Console-based text user interface with split-screen layout
- Menu panel on the left for main operations
- File browser panel on the right for file selection
- Status messages and dialog boxes for user feedback
- Keyboard-driven navigation and selection

**Sample Use Cases:**

1. **Encrypting a Sensitive Document**
   - User runs FileCryption
   - Navigates to their document using the file browser
   - Selects "Encrypt File" from the menu
   - Confirms file selection
   - Enters a strong password
   - Receives confirmation that the file was encrypted
2. **Decrypting a Protected File**
   - User runs FileCryption
   - Navigates to the encrypted file (.enc extension)
   - Selects "Decrypt File" from the menu
   - Confirms file selection
   - Enters the correct password
   - Receives confirmation that the file was decrypted successfully

---

## Individual member contribution:

**Vinaayak G Dasika: file_tree.h & file_tree.c**

**Yakshith Naidu: tui.h & tui.c**

**Y. Sai Aneesh Reddy: main.c**

**Yashita Anand: crypto.c & crypto.h**