Project Proposal: File Encryption Tool

by 67011619 and 67011596

Overview

This project aims to develop a **File Encryption Tool** using **AES-256-CBC encryption** in Rust. It will offer a user-friendly graphical interface for securely encrypting and decrypting files. Built on the principles of confidentiality and data security, it addresses the growing need for easy-to-use encryption software for personal and professional use. This tool will ensure that sensitive data can be stored and transferred securely, protecting it from unauthorized access or tampering. We will utilize **Rust** for its performance and security features, and the **Iced** framework for building the GUI, allowing cross-platform support.

Some examples where this tool might be useful:

1. A business team is collaborating on a project involving sensitive data. They can utilize this tool to encrypt their shared files, ensuring that only team members with the key can access the information. This way, they can work confidently, knowing their data is secure and can be decrypted when needed.
2. An individual is backing up personal files, like tax documents and private photos, to a cloud storage service. This tool allows them to encrypt these files before uploading, so even if the cloud storage is compromised, their sensitive information remains protected. They can easily decrypt the files whenever they want to access their data.
3. A lawyer needs to store sensitive client files containing legal documents. This tool can come to the rescue, allowing them to encrypt these records. They can easily decrypt the files whenever necessary using the keys, ensuring that confidential information remains protected from unauthorized access.

Project Goals:

1. **Ensure Strong File Security**: Implement AES-256 encryption to protect sensitive files.
2. **Provide a Simple User Interface**: Create an intuitive GUI for selecting, encrypting, and decrypting files.
3. **Automate Key and IV Generation**: Allow users to easily generate secure keys and initialization vectors.
4. **Enable Clipboard Integration**: Simplify key and IV management with copy-to-clipboard functionality.
5. **Give Clear User Feedback**: Display status messages to inform users of successful operations or errors.
6. **Aim for Cross-Platform Compatibility**: Design the tool in a way that it can be adapted for multiple operating systems (Windows, Linux, macOS) with minimal changes.

Core Features:

AES-256 Encryption with CBC Mode:

- Utilizes AES-256 for strong encryption, with CBC (Cipher Block Chaining) mode and PKCS7 padding, ensuring secure file encryption and decryption.

File Selection Interface:

- A user-friendly GUI allows users to select files for encryption or decryption via a file picker, streamlining the process.

Key and IV Management:

- Automatic generation of a secure encryption key and initialization vector (IV).
- Users can manually input keys and IVs or generate them within the tool, with validation for correct formats.

Clipboard Integration:

- One-click copy buttons allow users to copy the generated key and IV to the clipboard, facilitating users to store the keys somewhere safe for easy reuse.

File Encryption and Decryption:

- Supports both file encryption and decryption with feedback on success or failure.
- Encrypted files are saved with a .enc extension, and decrypted files are restored with <original name_decrypted.file_type>

Status Feedback:

- Clear status messages in the GUI inform the user about the current operation, errors, or success, enhancing usability.

Conclusion

This File Encryption Tool project aims to give users an easy and secure way to encrypt and decrypt files using strong AES-256-CBC encryption. Built with Rust and the Iced framework, the tool will be fast, dependable, and simple to use. It will enhance data security and privacy, and be designed to work across different platforms.