A
Minor Project Report
On

File Encryption and Decryption

Submitted in partial fulfillment of the requirements
For the award of the degree of

Bachelor of Technology
In
Computer Science and Engineering

By
Aastha Srivastava (CS-2341379)
Harsh Gupta (CS-2341625)
Shivang Singhal (CS-2341497)

Under the Supervision of
Dr. Gunjan Mittal Roy

School of Computer Science and Engineering

IILM University
Greater Noida, Uttar Pradesh
May, 2025

CERTIFICATE

This is to certify that the project report entitled "**File Encryption and Decryption Tool**" **submitted by**
Ms. Aastha Srivastava (CS-2341379)
Mr. Harsh Gupta (CS-2341625)
Mr. Shivang Singhal (CS-2341497)

to the IILM University, Greater Noida, Utter Pradesh in partial fulfillment for the award of Degree of Bachelor of Technology in Computer Science & Engineering is a Bonafide record of the minor project work carried out by them under my supervision during the year 2023-2024.

Dr. Gunjan Mittal Roy
Assistant Professor
School of CSE

Dr. Harshal Patil
Head of Dept.
Dept. of Computing & Security

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend my sincere thanks to all of them.

We are highly indebted to Dr. Gunjan Mittal Roy for her guidance and constant supervision. Also, we are highly thankful to them for providing necessary information regarding the project & also for their support in completing the project.

We are extremely indebted to Dr. Harshal Patil. We would also like to express our sincere thanks to all faculty and staff members of School of Computer Science and Engineering, for their support in completing this project on time.

We also express gratitude towards our parents for their kind co-operation and encouragement which helped me in completion of this project. Our thanks and appreciations also go to our friends in developing the project and all the people who have willingly helped me out with their abilities.

Aastha Srivastava
Harsh Gupta
Shivang Singhal

ABSTRACT

This minor project presents the design and implementation of an **Encryption and Decryption Tool**, a Python-based command-line utility for securing files using symmetric key cryptography. The system utilizes the Fernet module from the cryptography library to provide strong encryption and decryption for any file type, ensuring confidentiality and integrity. The tool features automatic key generation and management, robust error handling, and a simple user interface. It is designed for students and professionals who require an easy and effective way to protect sensitive files on their local systems. The project demonstrates practical application of cryptographic principles, addresses challenges such as key management and usability, and lays a foundation for future enhancements such as a graphical user interface and cloud integration.

KEYWORDS: *Encryption, Decryption, Fernet, Cryptography, File Security, Python, Symmetric Key*

CONTENTS

Chapter 1: Introduction

Data security is a critical concern in the digital era, with sensitive information frequently at risk of unauthorized access, theft, or tampering. Encryption is a proven method to safeguard data by converting it into an unreadable format, accessible only to those possessing the correct decryption key. The **Encryption and Decryption Tool** project aims to provide a robust, user-friendly solution for securing files using symmetric key cryptography. The tool leverages the Python cryptography library (Fernet module) to ensure confidentiality and integrity of user data through strong encryption algorithms. Designed as a command-line utility, it allows users to easily encrypt and decrypt files on their local system, making it suitable for students, professionals, and anyone concerned with data privacy.

SCOPE

This project delivers a command-line utility for encrypting and decrypting files using the Fernet symmetric encryption algorithm from the cryptography Python library. The tool is intended for personal use to protect sensitive files from unauthorized access.

Chapter 2: Literature Review

With the increasing concern over data privacy and cyber threats, secure file encryption has become a critical component of information security. Various cryptographic tools and libraries exist to ensure confidentiality, integrity, and authenticity of data. Among them, **Fernet**, a part of the cryptography library in Python, has gained popularity due to its simplicity, robustness, and adherence to modern encryption standards.

Overview of Encryption Technologies

Encryption technologies can be broadly categorized into:

- **Symmetric Encryption**, where the same key is used for both encryption and decryption.
- **Asymmetric Encryption**, involving a public-private key pair.

Fernet is a **symmetric encryption method** based on **AES in CBC mode with a 128-bit key**, **HMAC for authentication**, and **PKCS7 padding**. Key technologies and algorithms referenced in literature include:

- **AES (Advanced Encryption Standard)**: Known for its security and efficiency.
- **RSA**: A common asymmetric algorithm used for secure key exchange.
- **Blowfish, Twofish, DES**: Other symmetric encryption algorithms, each with varying levels of performance and security.

Comparative Studies

Several studies compare Fernet with other tools like:

- **PyCrypto** and **PyCryptodome**: Offer more flexibility but are lower-level.
- **GPG**: Provides stronger control and key management via asymmetric encryption but is more complex.
- **NaCl/Libsodium**: Preferred for highly secure messaging systems.

Fernet is preferred for ease-of-use and rapid prototyping, though it lacks advanced features like key rotation or multi-user encryption.

Chapter 3: Problem Statement

Despite the availability of encryption tools, many users do not secure their files due to complexity, lack of awareness, or fear of data loss if keys are misplaced. Existing solutions may be platform-specific, require advanced configuration, or lack robust error handling. There is a need for a lightweight, user-friendly, and secure encryption tool that simplifies the process of protecting sensitive files and manages keys safely, while providing clear feedback and error messages.

CHAPTER 4: PROPOSED WORK

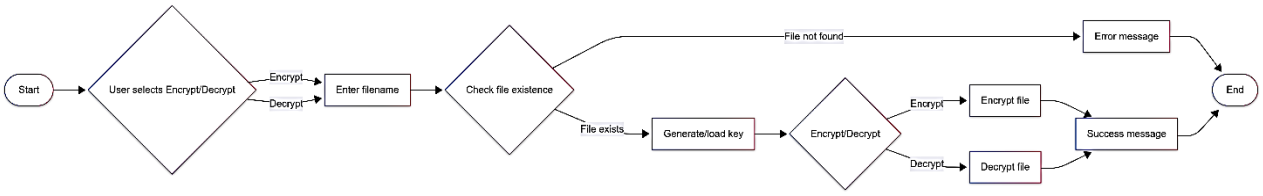
The proposed **Encryption and Decryption Tool** is a Python-based command-line application that enables users to encrypt and decrypt files securely using the Fernet symmetric encryption algorithm. The tool automatically generates and manages encryption keys, provides clear prompts and error messages, and ensures that only authorized users with the correct key can access encrypted files. The system overwrites original files with their encrypted or decrypted versions and is designed to be cross-platform and easy to use. Future enhancements may include a graphical user interface and cloud integration.

Features:

- Automatic key generation and secure storage
- File encryption and decryption
- Overwrites original files for security
- Comprehensive error handling (missing files, keys, invalid input)
- User-friendly prompts and messages

CHAPTER 5: SYSTEM DESIGN

5.1 Flow Chart



5.2 Data Flow Diagram (DFD)

- **External Entity:** User
- **Processes:** Input Choice, Input Filename, Key Management, Encryption/Decryption, Output Message
- **Data Stores:** File System, Key File (Secret.key)

5.3 Use Case Diagram

- **Actors:** User
- **Use Cases:** Encrypt File, Decrypt File, Generate Key, Display Error/Success

CHAPTER 6: IMPLEMENTATION

6.1 Technologies Used

- Python 3.x
- cryptography library (Fernet module)

6.2 Key Features

- Automatic key generation and secure storage (Secret.key)
- File encryption and decryption via command-line
- Robust error handling and user feedback

6.3 USER CHARACTERISTICS

• Users are expected to have basic familiarity with running Python scripts and using the command line.
• No prior cryptography knowledge is required.
• Users must have read/write permissions for the files they wish to encrypt or decrypt.

6.4 GENERAL CONSTRAINTS

• The tool operates only on files accessible from the local file system.
• The encryption key must be preserved; loss of the key results in permanent loss of access to encrypted files.
• The tool overwrites original files during encryption/decryption.
• Requires Python 3.x and the cryptography library.

CHAPTER 7: CONCLUSION, LIMITATION, AND FUTURE SCOPE

7.1 Conclusion

The Encryption and Decryption Tool successfully provides a secure, easy-to-use solution for file protection using symmetric key cryptography. It addresses the need for accessible encryption, robust error handling, and safe key management.

7.2 Limitations

- Loss of the key file (Secret.key) results in permanent loss of access to encrypted files.
- The tool overwrites original files (users must keep backups if needed).
- Currently operates only via command-line interface.

7.3 Future Scope

- Develop a graphical user interface (GUI) for broader accessibility.
- Add batch processing and password-based key management.
- Integrate with cloud storage for remote file encryption and backup.

7.4 Reference

- Cryptography Library Documentation
- Fernet Symmetric Encryption
- Python 3 Official Documentation
- [3041 Encryption Decryption Project, Scribd]
- [Base Encryption and Decryption Tool, JUIT]