CipherGuard

Prof. Sonawane A.A[1], Rutuja Bedre[2], Sarah Shaikh[3], Samiksha Muntode[4]
[1,2,3,4] *Department of Computer Engineering Sau. Sundarabai Manik Adsul Polytechnic, Chas, Ahilyanagar, Maharashtra*

***Abstract*—With the rapid growth of digital data, securing confidential files has become a critical concern. Unauthorized access, data theft, and file manipulation pose serious threats to personal and organizational information. This project presents a Secure File Encryption and Decryption System developed using Python and a graphical user interface (GUI) built with Tkinter.**

The system allows users to encrypt and decrypt files securely using cryptographic techniques, strong password validation, file-type restrictions, and encryption history tracking. The proposed system ensures data confidentiality, integrity, and controlled access while providing a user-friendly interface suitable for non-technical users

With the rapid expansion of digital information, protecting sensitive and confidential files has become a major challenge in today's digital environment. Unauthorized access, data leakage, and file tampering pose serious risks to both individuals and organizations. This project proposes a Secure File Encryption and Decryption System developed using Python with a user-friendly Graphical User Interface (GUI) implemented using Tkinter.

The system enables users to securely encrypt and decrypt files using strong cryptographic techniques. It incorporates strong password validation to prevent weak credentials, file-type restrictions to avoid unauthorized file usage, and an encryption history mechanism that records user activities along with date and time for auditing purposes. The application ensures that files remain inaccessible to unauthorized users and can only be decrypted using the correct encryption key.

The proposed system focuses on maintaining data confidentiality, integrity, and controlled access while offering a simple and intuitive interface suitable for non-technical users. This solution is effective for protecting personal and organizational data and can be further extended with advanced security features in the future

***Index Terms*—Cryptography, Fernet Algorithm, Python, File Security, Tkinter GUI, Data Encryption.**

I. INTRODUCTION

In today's digital era, sensitive information such as document, images, and reports is frequently stored and transferred electronically. Without proper security mechanisms, such data is vulnerable to unauthorized access and cyber threats. File encryption is one of the most effective methods to protect data by converting readable information into an unreadable format.

This project focuses on designing and implementing a secure file encryption and decryption system that enables users to protect files using encryption techniques. The system is developed using Python and provides a simple GUI to make encryption accessible

This project focuses on designing and implementing a secure file encryption and decryption system that enables users to protect files using encryption techniques. The system is developed using Python and provides a simple graphical user interface (GUI) to make encryption accessible even to non-technical users.

Traditional file protection methods such as basic passwords or file hiding techniques are often insufficient against modern cyber threats. To overcome these limitations, the proposed system uses cryptographic algorithms to ensure that files remain protected even if they are accessed by unauthorized users. Only users with the correct encryption key can decrypt and retrieve the original file.

Additionally, the system enforces strong password validation to prevent the use of weak passwords, thereby enhancing overall security. File-type restrictions are also implemented to allow only specific file formats, reducing the risk of misuse. An encryption history feature is included to track user activities such as file encryption and

decryption along with the date and time, which helps in monitoring and auditing.

Overall, this project aims to provide a secure, reliable, and user-friendly solution for protecting confidential files. The system can be effectively used in both personal and organizational environments where data security is a critical concern.

II. LITERATURE SURVEY

Several Existing File Security Systems Use Cryptographic Algorithms Such as Aes and Des to Protect Data. Traditional Tools Often Lack Usability, Proper Password Enforcement, Or Logging Mechanisms.

Some Systems Focus Only on Encryption Without Tracking User Actions, While Others Require Advanced Technical Knowledge. This Project Improves Upon Existing Systems.

By Integrating Strong Password Validation, File Type Restriction, Automatic Encryption History with Date and Time, And Simple Gui-Based Operation. The Proposed System Ensures That Encryption Is Accessible While Maintaining High Security Standards

Several Research Studies Have Highlighted the Importance of Data Security in File Storage and Transmission Systems. Many Existing File Encryption Applications Utilize Standard Cryptographic Algorithms Such as Aes and Des to Ensure Data Confidentiality.

However, Most Traditional Systems Are Complex, Command-Line Based, Or Lack a User-Friendly Interface, Making Them Difficult for Non-Technical Users.

In Addition, Many Existing Solutions Do Not Enforce Strong Password Policies, Which Can Lead to Security Vulnerabilities.

Some Systems Provide Encryption Functionality but Fail to Maintain Logs or History of User Activities, Making It Difficult to Track File Operations and Identify Unauthorized Access.

The Proposed System Addresses These Limitations by Combining Strong Password Validation, File Type Restrictions, And Automatic Encryption History Tracking with Date and Time

By Incorporating a Simple Gui-Based Approach Using Python and Tkinter, The System Improves Usability While Maintaining High Security Standards, Making It Suitable For Both Personal And Organizational Use.

III. PROPOSED METHODOLOGY

The proposed system is a desktop-based file encryption and decryption application developed using Python. It allows users to select files, apply encryption using secure cryptographic methods, and decrypt files using the correct key. The system also records encryption and decryption activities for transparency and tracking.

the proposed system is a desktop-based file encryption and decryption application developed using the python programming language with a graphical user interface built using tkinter. the system is designed to provide secure, simple, and controlled access to file encryption for end users.

first, the user selects a file from the local system through the gui. the application performs file type validation to ensure that only allowed file formats such as pdf, jpg, png, and docx are processed. this helps prevent unsupported or malicious files from being encrypted.

for encryption, the user is required to enter a strong password. the system validates the password based on predefined security rules such as minimum length and character requirements. after successful validation, the file is encrypted using a secure cryptographic algorithm and stored in encrypted format. a unique encryption key is generated and required for decryption.

for decryption, the user selects the encrypted file and provides the correct key. if the key is valid, the system restores the original file. if an incorrect key is entered, access is denied, ensuring data confidentiality.

the system also maintains an encryption history log that records the file name, operation type (encryption or decryption), date, and time. this feature improves transparency, tracking, and accountability. overall, the proposed methodology ensures data security, integrity, and user-friendly operation.

A. Key Features

1. Secure file encryption and decryption
2. Strong password validation

3. File type restriction
4. Encryption history with date and time
5. User-friendly GUI

B. System Workflow The system follows these steps:

1. User selects a valid file (PDF, JPG, PNG, DOCX)
2. Password strength is verified
3. File is encrypted using a cryptographic algorithm
4. Encryption key is generated and provided to the user
5. Decryption requires the correct key
6. All actions are logged in an encryption history file

IV. RESULTS AND DISCUSSION

The system successfully encrypts and decrypts files while preventing unauthorized access. Password validation ensures stronger security, and file restriction prevents unsupported file types. The encryption history feature records each action with file name, date, time, and operation type, improving accountability.

The developed system successfully encrypts and decrypts files while ensuring protection against unauthorized access. The encryption mechanism converts the original file into an unreadable format, making it inaccessible to unauthorized users, while the decryption process restores the file only when a valid encryption key is provided.

The implementation of strong password validation significantly enhances system security by preventing the use of weak or easily guessable passwords. This reduces the risk of brute-force and unauthorized access attacks. Additionally, file type restriction ensures that only supported file formats are processed, preventing misuse and enhancing system reliability.

The encryption history feature records every encryption and decryption operation along with the file name, date, time, and type of action performed. This logging mechanism improves transparency and accountability by allowing users to track file activities effectively. Overall, the results demonstrate that the proposed system provides a secure, reliable, and user-friendly solution for file protection.

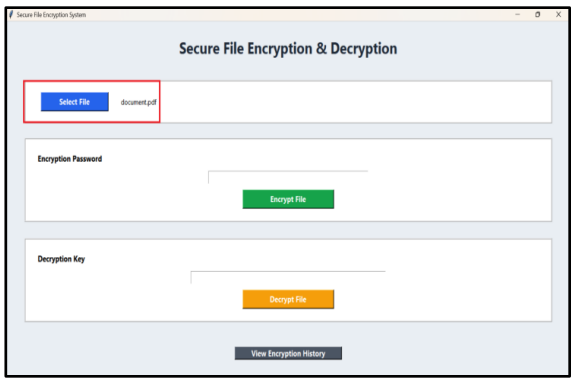

Fig. 1. File Selection Interface

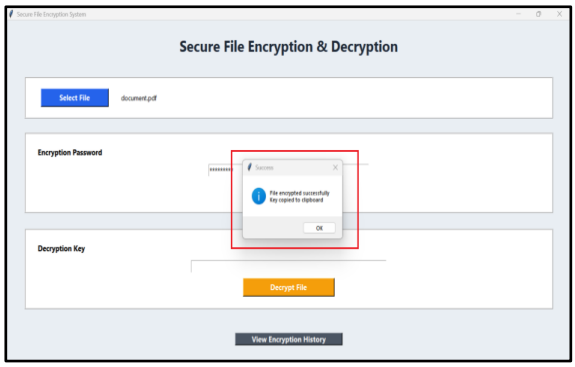

Fig. 2. Successful File Encryption

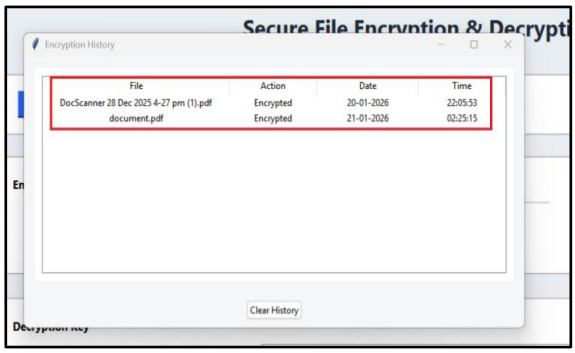

Fig. 3. Encryption History with Date and Time

V. CONCLUSION

This project successfully demonstrates a secure and user-friendly file encryption and decryption system. By combining cryptographic techniques with a simple GUI, the system provides effective data protection for everyday users. The inclusion of password validation, file restrictions, and encryption history enhances both scurity and usability

This project successfully demonstrates the design and implementation of a secure and user-friendly file encryption and decryption system. By integrating

strong cryptographic techniques with a simple and intuitive graphical user interface, the system provides effective data protection for everyday users as well as organizations.

The inclusion of strong password validation, file type restrictions, and an encryption history feature enhances both security and usability. These features ensure controlled access, prevent unauthorized file operations, and allow users to monitor encryption and decryption activities efficiently. Overall, the system proves to be a reliable solution for protecting sensitive digital data in a secure and accessible manner.

VI. FUTURE SCOPE

in the future, the proposed system can be enhanced by integrating advanced security features and additional functionalities. cloud storage support can be added to allow secure encryption and decryption of files stored on cloud platforms. this will make the system more flexible and accessible.

the system can also be extended to support multiple encryption algorithms, allowing users to choose the level of security based on their requirements. implementing user authentication and role-based access control can further strengthen system security.

additional improvements may include batch file encryption, mobile application support, and automatic key management mechanisms. these enhancements will increase scalability, usability, and applicability of the system in real-world environments.

REFERENCES

[1] William Stallings, Cryptography and Network Security, Pearson Education.

[2] Python Software Foundation, "Python Official Documentation," [Online]. Available: https://docs.python.org/3/.

[3] National Institute of Standards and Technology (NIST), "Guidelines for Using Cryptographic Standards," Special Publication.

[4] Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed., John Wiley & Sons.

[5] Atul Kahate, *Cryptography and Network Security*, Tata McGraw-Hill Education.

[6] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press.

[7] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126.

[8] N. Ferguson, B. Schneier, and T. Kohno, *Cryptography Engineering*, Wiley Publishing.

[9] Kahn Academy, *Introduction to Cryptography*. [Online]. Available: https://www.khanacademy.org/computing/computer-science/cryptography

[10] IEEE Xplore Digital Library, *Research Papers on File Encryption and Data Security*. [Online].