Project:- Develop a Simple File Encription and Decryption Tools Using AES
(Criptography & Network security)
Computer Science Application

A6 Project

Session: 2024-2025

Project Guide:
Narayan Vyas

Submitted By:
Akash Kumar(23CSA2BC269)
Niranjan das (23CSA2BC233)
Md.Nafis Siddique(23CSA2BC213)
Suraj kumar(23CSA2BC188)
Akansha Kumari(23CSA2BC219)

Certified that this project report **"Develop a Simple File Encription and Decryption Tool Using AES "** is the bonafide work of "**Niranjan Das , Akash kumar , Md. Naffis Siddiqe , Suraj kumar, Akansha kumari** who carried out the project work under my/our supervision.

TABLE OF CONTENTS

ABSTRACT

Cryptography is used to secure and protect data during communication. It is helpful to prevent unauthorized person or group of users from accessing any confidential data. Encryption and decryption are the two essential functionalities of cryptography [4].A message sent over the network is transformed into an unrecognizable encrypted message known as data encryption. At the receiving end, the received message is converted to its original form known as decryption [2].Encryption is a process which transforms the original information into an unrecognizable form. This new form of the message is entirely different from the original message. That's why a hacker is not able to read the data as senders use an encryption algorithm. Encryption is usually done using key algorithms. Decryption is a process of converting encoded/encrypted data in a form that is readable and understood by a human or a computer. This method is performed by un-encrypting the text manually or by using keys used to encrypt the original data.

CHAPTER 1: INTRODUCTION

1.1 Identification of Client/Need/Relevant Contemporary Issue

In the modern digital era, the exponential growth of data and its extensive use in various fields, including business, healthcare, finance, and personal communication, have heightened the importance of data security. As organizations and individuals increasingly rely on digital systems for data storage and transmission, they face significant challenges related to data protection from unauthorized access, theft, and cyber-attacks. Consequently, there is an urgent need for robust encryption mechanisms to ensure data confidentiality and integrity.

One of the most reliable encryption algorithms in use today is the Advanced Encryption Standard (AES), which is recognized for its security, efficiency, and widespread adoption in various applications. AES encryption is crucial for safeguarding sensitive data from both passive and active attacks, making it indispensable for secure communication, file storage, and data transmission. However, despite its importance, many users, especially those with limited technical expertise, face challenges in utilizing AES-based encryption tools effectively. To address this contemporary issue, a simple yet efficient file encryption and decryption tool using AES is proposed, focusing on usability and security.

1.2 Identification of Problem

The rapid expansion of digital infrastructure and increased connectivity have created new avenues for cyber threats, including data breaches, unauthorized access, and data tampering. Sensitive information, whether stored locally or transmitted over networks, is increasingly at risk from malicious actors seeking to exploit vulnerabilities. Traditional data storage methods that do not incorporate encryption expose valuable information to theft and manipulation.

Moreover, although AES encryption offers strong protection, existing tools and software often lack user-friendliness, making it difficult for non-technical users to secure their files effectively. As a result, many individuals and small organizations are hesitant to adopt encryption practices, leaving their data exposed to potential security breaches. The fundamental problem lies in the absence of a comprehensive, easy-to-use encryption and decryption tool that leverages AES technology to protect digital assets.

1.3 Identification of Tasks

To develop a comprehensive and user-friendly file encryption and decryption tool using AES, the following tasks have been identified:

*Conduct an extensive review of AES encryption standards, including algorithm principles and implementation techniques.

*Design a graphical user interface (GUI) that simplifies the process of file encryption and decryption for users with varying levels of technical expertise.

*Implement AES encryption and decryption functionalities that support a wide range of file formats and sizes.

*Integrate robust error handling, performance optimization, and data integrity checks to ensure the tool's reliability.

*Perform thorough testing to evaluate the tool's performance, security robustness, and usability under different scenarios.

*Develop comprehensive documentation and user guides to facilitate adoption and usage by non-technical users.

*Gather feedback from potential users and stakeholders to refine and improve the tool’s features and usability

CHAPTER 2: LITERATURE REVIEW/BACKGROUND STUDY

2.1 Existing Solutions

In the field of data security, numerous encryption tools and algorithms have been developed to protect digital assets. Among these, AES-based encryption tools are widely used for their robustness and efficiency. Some popular solutions include VeraCrypt, AxCrypt, and BitLocker. VeraCrypt is an open-source disk encryption tool that enhances security but is primarily focused on full disk encryption rather than individual file encryption. AxCrypt is designed for file-level encryption and is user-friendly but lacks advanced customization options. BitLocker, integrated with Windows, offers full-disk encryption but does not specifically address file-based encryption needs.

Despite the availability of these tools, many users find them overly complex or limited in scope. Additionally, most existing solutions do not effectively balance usability with strong encryption. Hence, there remains a demand for a tool that combines AES encryption with simplicity and versatility, addressing both technical and non-technical users’ needs.

2.2 Problem Definition

Existing encryption solutions either focus on full-disk encryption or are challenging for non-technical users to operate. As a result, many individuals and small businesses avoid using encryption altogether, leaving sensitive data at risk. Furthermore, some existing tools are platform-dependent or lack support for a wide range of file formats. The primary problem is the lack of a cross-platform, user-friendly file encryption tool that leverages AES while being accessible to users with minimal technical skills.

2.3 Goals/Objectives

The main goal of this project is to develop a simple, efficient, and secure file encryption and decryption tool using AES. The tool should be user-friendly, allowing users with minimal technical knowledge to encrypt and decrypt files seamlessly. Specific objectives include:

*Implementing robust AES encryption and decryption algorithms.

*Designing an intuitive graphical user interface (GUI).

*Ensuring cross-platform compatibility (Windows, Linux, and MacOS).

*Supporting various file formats and sizes.

*Integrating performance optimization and error handling mechanisms.

*Conducting rigorous testing to ensure security and reliability.

*Providing user documentation and technical support resources.

CHAPTER 3: RESULTS ANALYSIS AND VALIDATION

Implementation of Solution

The implementation of the file encryption and decryption tool using AES involved several phases. The initial phase focused on researching the AES algorithm to ensure accurate implementation. Next, the development of the tool began with designing a user-friendly graphical interface, allowing users to select files for encryption or decryption easily. The tool was developed using Python with libraries like PyCryptodome for encryption and Tkinter for GUI design.

The encryption process involved generating a secure AES key and initializing it with a unique initialization vector (IV) for each file. The tool then used the AES algorithm in CBC mode to encrypt the file, ensuring confidentiality and data integrity. Similarly, the decryption process reversed the encryption steps using the same key and IV. Error handling mechanisms were incorporated to detect incorrect keys or corrupted files, preventing data loss.

CHAPTER 4: IMPLEMENTATION AND METHODOLOGY

4.1 Data Collection and Preprocessing

The development of the tool involved collecting diverse file formats for testing, including text, image, audio, and document files. The data was preprocessed to ensure compatibility with the encryption and decryption algorithms, and appropriate file handling methods were incorporated.

4.2 Feature Selection

Key features selected for implementation included encryption, decryption, key generation, and error handling. These features were chosen to maximize security while maintaining user-friendliness.

4.3 Model Selection and Training

The model involved selecting the AES encryption algorithm with a secure key length (256 bits). Implementation was carried out using the PyCryptodome library to ensure high performance and robust encryption.

4.4 Performance Evaluation

Performance evaluation was conducted using various file sizes and formats to assess the encryption and decryption speeds. The tool demonstrated low latency and high throughput, ensuring efficient performance in real-world scenarios.

CHAPTER 5: RESULTS AND DISCUSSION

Analysis of Results

The AES-based file encryption and decryption tool was subjected to a series of rigorous tests to evaluate its performance, usability, and security. Various file formats, including text, images, audio, and

documents, were encrypted and decrypted using the tool. The testing focused on measuring encryption and decryption speeds, CPU and memory usage, and the tool's ability to handle files of different sizes.

Results indicated that the tool performed efficiently with minimal computational overhead. For small to medium-sized files (up to 100 MB), the encryption and decryption processes were almost instantaneous. For larger files, the performance remained satisfactory, with slight increases in processing time corresponding to file size growth. CPU and memory consumption remained within acceptable limits, demonstrating the tool's optimization and efficiency.

Security analysis was conducted to assess the robustness of the encryption method. The AES algorithm in CBC mode was found to effectively prevent common cryptographic attacks, including brute force and differential cryptanalysis. Additionally, error handling mechanisms were tested by intentionally introducing corrupt data and incorrect keys, which the tool accurately identified and reported without compromising data integrity.

CHAPTER 6: CONCLUSION AND FUTURE WORK

Conclusion

The development of the AES-based file encryption and decryption tool has addressed the critical need for secure and efficient data protection. With the rapid growth of digital data, ensuring data confidentiality and integrity has become paramount. Through this project, a practical and user-friendly tool has been created that leverages the Advanced Encryption Standard (AES) to safeguard sensitive information from unauthorized access and potential cyber threats.

The tool's design prioritized simplicity and usability while maintaining robust encryption standards. By employing AES in CBC mode, the tool achieved strong security against common cryptographic attacks. Comprehensive testing demonstrated the tool's effectiveness in encrypting and decrypting various file formats with minimal computational overhead, making it suitable for both technical and non-technical users.

Future Work

To further enhance the tool's capabilities and usability, several future improvements are proposed:

*Batch Encryption and Decryption: Implementing support for processing multiple files simultaneously to improve efficiency.

*Advanced Settings: Allowing users to customize AES key length and choose encryption modes for greater flexibility.

*Enhanced File Format Support: Extending compatibility to include more proprietary and less common file formats.

*Multi-Language Support: Adding language localization to increase accessibility for non-English-speaking users.

*Cloud Integration: Enabling encryption and decryption of files stored in cloud services, ensuring data security beyond local storage.

*Improved Key Management: Incorporating secure key storage and retrieval mechanisms to enhance user convenience and safety.

REFERENCES

Daemen, J., & Rijmen, V. (2002). The Design of Rijndael: AES - The Advanced Encryption Standard. Springer-Verlag.

Stallings, W. (2017). Cryptography and Network Security: Principles and Practice (7th Edition). Pearson.

Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C. Wiley.

National Institute of Standards and Technology (NIST). (2001). FIPS PUB 197: Advanced Encryption Standard (AES).

Ferguson, N., & Schneier, B. (2003). Practical Cryptography. Wiley.

PyCryptodome Documentation. (n.d.). Retrieved from https://pycryptodome.readthedocs.io/

Tkinter GUI Programming by Example. (2018). Packt Publishing.