 DOI:10.5281/zenodo.17795962

Title

ENCRYPTION AND DECRYPTION SYSTEM FOR SECURE FILE PROTECTION

Author

BENJAMIN MAKWANGWALA

Co-Author

MR JOEL MULEPA

Issue August 2025

Certificate AR2025GAT3R

ABSTRACT

This study presents the design and implementation of a software system capable of encrypting and decrypting multiple file formats—specifically documents, text, MP3s, MP4s, and images. In today's digital age, the security of personal and confidential data is of utmost importance. Unauthorized access, data theft, and cyber-attacks threaten information stored on digital devices. This project addresses these challenges by developing a simple, user-friendly file encryption and decryption tool for PCs. The methodology involves using cryptographic algorithms to transform readable files into encrypted formats and restore them back when needed. The system was tested on various file types to evaluate its performance, effectiveness, and reliability. Results show that the system efficiently encrypts and decrypts files without data loss, ensuring confidentiality and protection. This tool is useful for individuals, organizations, and institutions needing a secure method of storing and transferring files. Future improvements could include cross-platform support and integration of advanced encryption standards.

KEYWORDS: Encryption, Decryption, Data Security, File Protection, Cryptography, Information Privacy

INTRODUCTION

Background of Study

In an increasingly digital world, the need for protecting sensitive data is more urgent than ever. Files such as documents, images, audio, and video are frequently transmitted and stored on personal and organizational systems. Without adequate protection, this data can be intercepted, modified, or misused. Encryption and decryption are essential mechanisms in data security. Encryption transforms readable information into a coded format, while decryption restores it to its original form. This project aims to build a system that encrypts and decrypts common file types—text, MP3, MP4, images, and documents—on a PC environment. The tool is intended to offer simple and reliable protection to digital files, especially for non-technical users. The motivation behind this project stems from growing concerns over data privacy and the lack of accessible security tools for everyday users.

Objectives

The primary goal of this research is to develop and evaluate an Encryption and Decryption System capable of securely protecting multiple file formats using cryptographic techniques. The study is guided by the following specific objectives:

1. To promote data security and privacy preservation: This objective emphasizes

the social and practical impact of the system. Many files in Malawi and beyond are vulnerable due to lack of encryption tools. By integrating these formats into a secure platform, the project helps safeguard digital privacy and ensures that users can protect their data without barriers, promoting confidentiality and inclusion. Moreover, it supports individuals and organizations in securing and managing files for privacy and archival purposes.

2. To enhance accessibility and user interaction through secure technology: This objective focuses on usability and inclusiveness. The project aims to build a platform that enables users, especially those with limited technical skills, to interact easily with security tools through a graphical interface. Instead of complex commands, users can encrypt or decrypt files with a few clicks. This improves access to security features, especially in rural or underserved communities. It also benefits individuals with varying tech literacy by offering a more natural and convenient way of file protection through simple interaction.
3. To design an efficient and adaptable cryptographic model for real-time file processing: The goal is to build a robust model that performs encryption and decryption quickly and accurately, even with varying file sizes or formats. The model should also be adaptable, meaning it can be extended to support additional formats or standards as more needs arise. Real-time processing capability is essential to ensure smooth and instant operations during file handling, transfers, or storage. Ultimately, this will make the system reliable, efficient, and scalable for broader applications across devices and file types.

LITERATURE REVIEW

A literature review serves as a critical analysis of existing scholarly works relevant to a particular topic or research question. It provides a comprehensive overview of the current state of knowledge, identifies gaps, and synthesizes key findings to inform further research. By examining and synthesizing existing literature, researchers gain insights, contextualize their own work, and contribute to the advancement of knowledge in the field.

Overview of Research Studies

A wide body of research has focused on encryption algorithms and their application in data security. Traditional techniques such as the Data Encryption Standard (DES), Advanced Encryption Standard (AES), and RSA have been studied and implemented in various security

systems. Researchers like Schneier (1996) emphasized the importance of strong encryption in digital communication, while Stallings (2005) explored cryptographic principles in real-world applications. More recent works address the usability of encryption tools and their integration into user-friendly platforms. Studies also reveal gaps in availability of simple encryption systems for common users, especially those not in enterprise environments. This project builds upon existing encryption research, focusing on practical file protection rather than complex network security.

METHODOLOGY AND TOOLS

This study employed a Design Science Research (DSR) methodology, which emphasizes the creation, testing, and refinement of innovative technological artifacts to solve real-world problems. In this context, the key challenge addressed is the limited availability of user-friendly encryption systems capable of handling multiple file types, especially within low-resource settings like Malawi.

The DSR framework was suitable as it integrates both scientific rigor and practical innovation, enabling a structured yet adaptable process for designing and evaluating a cryptography-based file protection application. The methodology followed three major phases: system design, system development, and system evaluation.

Each phase was guided by the agile methodology, which supports iterative development, rapid prototyping, user feedback, and continuous system improvement. Agile divides the development cycle into short, manageable sprints, ensuring that user input and real-world testing inform every iteration of the system.

System Design Phase

The design phase began with the identification of both functional and non-functional requirements. Data collection involved literature review and user surveys to understand file diversity, format variations, and user expectations.

The system's architecture was then conceptualized, focusing on modularity, scalability, and adaptability to multiple file types. The design process emphasized:

- A file acquisition module to capture and preprocess input;
- A cryptography-based engine for format-specific encryption and decryption; and
- A user interface for real-time operation output.

The design also included structures for storing keys, files, and results. Furthermore, security experts were engaged to help identify format variations and algorithms commonly used in

target setups. This ensured practical and technical relevance of the system.

System Development Phase

The development phase involved implementing the designed architecture into a functional prototype. Development was conducted in Agile sprints, where each sprint targeted specific components such as file preprocessing, encryption modeling, decryption generation, and graphical user interface (GUI) creation.

Key tools used in this phase included Python and third-party cryptographic libraries for algorithm implementation and file handling, while Tkinter was used for interface management. The encryption-to-decryption pipeline integrated AES models fine-tuned for various file formats.

Each sprint ended with functional testing, where developers validated the accuracy of encryption and decryption output. Feedback from users was incorporated before moving to the next sprint, promoting an iterative and user-centered design process.

System Evaluation Phase

In the evaluation phase, the prototype system was tested in a controlled environment involving participants familiar with file handling. A pilot test was conducted, where participants processed various files, and outputs were analyzed for accuracy, latency, and reliability.

Evaluation metrics included:

- Encryption/Decryption Success Rate for processing accuracy;
- Response Time for real-time operations; and
- User Satisfaction Scores through post-test surveys.

The system achieved high accuracy with stable performance in real-time operation. Ethical considerations such as informed consent, data anonymization, and participant privacy were strictly maintained throughout the testing process.

Justification for Agile Methodology

The Agile methodology was adopted due to its adaptability, focus on user collaboration, and iterative improvement cycle. Unlike traditional waterfall models, Agile allowed developers to respond rapidly to challenges, such as differences in file formats or processing clarity.

Frequent feedback from users, experts, and technical evaluators ensured that modifications

were implemented promptly without disrupting the entire workflow. This approach minimized development risks, improved system usability, and enhanced stakeholder engagement—all of which are critical for a technology-sensitive system that evolves with user interaction and file diversity.

Development Tools

The implementation of the Encryption and Decryption System required a combination of programming languages, frameworks, and tools to enable robust processing and secure data handling.

System Architecture

The development of the Encryption and Decryption System relied on a combination of backend, frontend, and auxiliary tools to ensure efficiency, accuracy, and scalability. The backend tools used were Python and cryptographic libraries like cryptography or PyCrypto, which handled the system's logic, key management, and secure data processing. MySQL or file-based storage provided a reliable environment for storing encrypted data and keys. The frontend tools included Tkinter or similar GUI libraries, which were used to design a responsive and user-friendly interface, enabling users to select files and view processing outputs seamlessly across operations. For scripting and algorithm development, Python served as the core programming language, supporting preprocessing, encryption, and feature extraction using libraries such as os and shutil for file handling. In addition, tools like buffer optimization were utilized for large files, including handling and normalization. Together, these tools created a robust and integrated technological foundation that ensured the system's ability to securely process files while maintaining efficiency, data integrity, and user accessibility.

Data Collection and Preprocessing Data Sources

The Encryption and Decryption System relies heavily on the quality and diversity of its test files. Datasets were collected from a combination of open-source repositories and locally created samples to ensure high processing accuracy and format relevance. The project adopted a multi-layered approach to curate, preprocess, and validate all files used to test the cryptographic model.

Open-Source Datasets: Foundational files for the system were obtained from publicly available sources such as sample media libraries. These provided general examples across different formats and sizes. To ensure the system could handle local variations, additional samples were

created or collected from users in community settings.

Locally Curated and Expert-Reviewed Data

To ensure accuracy, locally created files were reviewed by experts and users. The reviewers verified format consistency, integrity, and distinctions for files with varying structures. Their contributions were essential in eliminating errors and ensuring that the system reflected authentic file patterns.

This collaborative process helped enhance the model's reliability, appropriateness, and depth.

Data Cleaning and Noise Filtering

Before testing, all collected files underwent rigorous preprocessing. Samples were cleaned using tools to remove corruption, normalize sizes, and ensure clear input. The files were segmented into uniform types for easier processing.

Details were standardized to remove inconsistencies and irregular formats. This improved data quality and helped the cryptographic engine learn precise relationships between input and output.

Language and Tone Filtering

Because files may include textual content with variations, classification tools were integrated during preprocessing for text-based files. Each sample was labeled based on type (e.g., plain text, embedded). This step was critical in helping the model distinguish between similar files that vary by content.

Localization and Language Support

Considering diversity, the system was designed to handle multilingual text input, primarily focusing on English and local variations. Surveys and reviews were conducted to collect expressions and region-specific formats. This localization ensured that the model was contextually accurate, sensitive, and adaptable to local file handling styles.

Testing and Evaluation Study Design

To evaluate system performance, a pilot study was conducted involving participants who were users of various file types. Participants processed short files, which were then handled by the system. The resulting outputs were compared against original references. Feedback was collected through surveys to assess usability, speed, and accuracy of the system.

Types of Testing Performed

- Usability Testing: Examined how intuitive and accessible the application

interface was. Participants evaluated ease of selecting files, processing, and output viewing.

- Functional Testing: Verified whether the main functions including input, saving, processing generation, and export worked correctly under different use conditions.
- Accuracy Testing: Measured how precisely the system encrypted and decrypted files. Results were compared with originals to determine the Error Rate and Processing Accuracy.
- Performance and Reliability Testing: Evaluated response time, stability, and system performance under varied workloads.
- Security and Data Handling Testing: Ensured data confidentiality through encryption, secure key management, and integrity of processed files, in compliance with ethical standards.

Evaluation Metrics

The project evaluated several performance indicators:

- Processing Accuracy: Degree of precision between encrypted/decrypted and original files.
- Ease of Use: Simplicity and navigability of the interface.
- Processing Speed: Time taken to convert input to output.
- System Reliability: Uptime, crash rates, and response times.
- Relevance: Effectiveness in handling local file formats.

Ethical Considerations

All participants gave informed consent prior to testing. The project adhered to strict data protection and ethical research standards. No personal data was stored; all samples were anonymized and secured. Participants were fully informed that their data would be used solely for research and system improvement purposes.

RESULTS

The results of the Encryption and Decryption System were evaluated and analyzed based on three key dimensions: system performance, user experience, and technological impact. These dimensions provide a holistic understanding of the system's effectiveness, usability, and contribution to data security advancement.

System Performance

The first dimension focused on the technical accuracy and efficiency of the cryptographic model. The system was tested using multiple file

samples collected from diverse formats differing in size and type. The model demonstrated high processing accuracy, which significantly improved after optimization and buffer techniques.

Processing time was found to be efficient, allowing near real-time operations. Furthermore, the system successfully handled variations and sizes, achieving reliable results even in large inputs. The integration of algorithms such as AES contributed to higher accuracy in continuous processing. These findings affirm that the system performs effectively in real-world environments and can be optimized further through additional expansions.

User Experience

Evaluated usability, accessibility, and user satisfaction. Field testing was conducted with participants from local communities, professionals, and students. Feedback revealed that users found the system intuitive, responsive, and user-friendly, especially those with limited technical skills.

The interface-based interaction allowed users to secure files naturally without switching to complex tools. This created a sense of inclusion among users. Additionally, the output was clear, and processing accuracy built trust in the system's capability. The system's interface also enabled users to toggle between modes, promoting flexibility. Overall, users rated the system as helpful and easy to use, confirming its practical value in everyday settings.

Technological and Societal Impact

This focused on examining the broader technological relevance and social contribution of the project. The introduction of cryptography-powered protection for files represents a major step toward inclusivity and digital equity. The system not only bridges the gap between technology and users but also preserves privacy through secure means.

From a technological standpoint, the project demonstrated the feasibility of low-resource cryptography development—a challenge often faced by underrepresented setups. The successful implementation using limited resources proves that standard algorithms can overcome constraints. Furthermore, the system has potential applications in education, business, and governance, where file protection can enhance data accessibility and user engagement.

In essence, the project's impact goes beyond technology—it empowers populations to secure, store, and share digitally, contributing to sustainable digital transformation.

Discussion

The findings underscore the potential of cryptography-based systems in promoting

privacy and accessibility. High usability and accuracy scores demonstrate that with proper preparation and tuning, files can be effectively secured. Compared with conventional tools, this model performed better in handling multimedia and expressions, making it more relatable to users.

Nevertheless, some limitations such as challenges in large file environments and cross-platform support highlight the need for more diverse testing and improvements. The system represents a significant step toward bridging the gap between technology and users.

CONCLUSION

This study successfully developed and evaluated an Encryption and Decryption System tailored for diverse file formats. The system demonstrated high levels of accuracy, responsiveness, and usability. By leveraging cryptography and secure technologies tested on various files, the project contributes to digital privacy, professional advancement, and improved accessibility. Future improvements will focus on expanding format coverage, integrating mobile deployment, and refining the model for real-time biometric and cloud support.

The project confirms that security systems for files can serve as powerful tools for inclusion, protection, and enhancement across Africa and beyond.

REFERENCES

Schneier, B. (1996). Applied Cryptography. John Wiley & Sons.

Stallings, W. (2005). Cryptography and Network Security. Pearson Education.

Singh, S. (1999). The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography. Anchor.

National Institute of Standards and Technology (NIST). (2001). Advanced Encryption Standard (AES). [FIPS Publication 197].

Paar, C., & Pelzl, J. (2009). Understanding Cryptography: A Textbook for Students and Practitioners. Springer.