International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 07 July-2026 | Impact Factor: 3.5

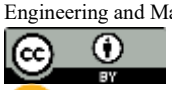
Advanced Secure File Storage and Sharing System using Hybrid Cryptography and Cloud Computing Integration

Mr. Dhanna Ram[1]
Lecturer, Department of Computer Science and Engineering
Government Polytechnic College,
Churu, Rajasthan, India
dr.mecs2009@gmail.com

Mr. Mohit kumar saini[2]
Lecturer, Department of Computer Science and Engineering
Government Polytechnic College,
Churu, Rajasthan, India
1saini1mohit1@gmail.com

Pooja Verma[3]
Lecturer, Department of Computer Science and Engineering
Government Polytechnic College,
Churu, Rajasthan, India
poojav074@gmail.com

How to Cite this Article:
Ram, D., saini, M. K. & Verma, P. (2026). Advanced Secure File Storage and Sharing System using Hybrid Cryptography and Cloud Computing Integration. International Journal of Creative and Open Research in Engineering and Management, 02(7). https://doi.org/10.55041/ijcope.v2i7.026



https://doi.org/10.55041/ijcope.v1i1.001

Abstract- **In the contemporary digital era, the necessity for safe and efficient file storage has grown critical, particularly in cloud computing contexts. The work proposes the creation of a File Storage System that utilises the scalability and accessibility of cloud computing while improving data security and privacy through hybrid cryptography methods. This new system seeks to deliver comprehensive encryption, efficient data transfer, and dependable data integrity verification for individuals and organisations alike. The technology will serve anyone seeking to protect their data while utilising cloud-based services. The amalgamation of hybrid cryptography and cloud computing will provide a scalable and secure solution, establishing this project as a prospective transformative force in cloud-based file storage and sharing.**

Keywords— ***Secure File Storage, Cryptography, Hybrid Cryptography***

1. Introduction

Safeguarding sensitive information during cloud storage is crucial in the current era of incessant cyberattacks and data breaches. Hybrid cryptography is an effective technique for safeguarding sensitive data during transmission or storage, as it integrates the benefits of both symmetric and asymmetric encryption. Developers can create a secure file storage technique that encrypts data locally prior to transmitting it to the cloud by utilizing Python's comprehensive cryptography libraries.

Acquire knowledge about utilizing Python and the concept of hybrid cryptography to safeguard your cloud-based files in this comprehensive essay. We will examine the fundamentals of symmetric and asymmetric encryption, discuss the advantages of hybrid cryptography, and demonstrate the application of Python's cryptography package to implement these concepts. Exploring this topic will reveal how to create a robust and secured file storage system that protects the integrity and confidentiality of data stored in cloud environments. The transition to cloud computing has fundamentally transformed how businesses manage and access their data in the contemporary digital landscape. However, this transition introduces more security problems, as data transmitted and stored on remote servers may be accessible to unauthorized entities. Encryption has become essential for protecting data stored in cloud systems, thereby alleviating these threats.

Hybrid cryptography, which integrates symmetric and asymmetric encryption techniques, can significantly enhance data security both at rest and in transit. For bulk data encryption, symmetric encryption is preferable as it necessitates only a single key for execution. Conversely, asymmetric encryption employs a public key for encryption and a private key for decryption, thereby enhancing the security of digital signatures and key exchanges.

This essay explores the concept of secure cloud file storage via Python-based hybrid cryptography. We analyze the merits and drawbacks of symmetric and asymmetric encryption by examining their foundational principles. Utilizing the cryptography package in Python, we demonstrate the local symmetric encryption of files, the asymmetric encryption of the symmetric key, and the secure storage of the encrypted content in cloud storage.

Throughout our analysis, we emphasize the advantages of hybrid cryptography, including robust data protection and efficient key management. Key creation, encryption methodologies, and secure data transmission protocols are critical factors to consider when establishing secure file storage systems in practical applications. This guide enables developers to master the intricacies of constructing a secure file storage system that safeguards sensitive data in the cloud from hackers and other malevolent entities.

1.1 Problem Statement

Cloud computing has revolutionized how people and business keep, use and share data. The use of third party cloud servers does, however, present numerous security issues such as unauthorised access, data breaches, privacy concerns and data tampering. Standard security methods typically use just one encryption method and may not be able to protect against today's threats. In addition, users often struggle to share encrypted files securely, ensuring the confidentiality, integrity, and access control of their data. Current cloud storage options are also inadequate in meeting security, scalability, usability and cost requirements. Thus, it is essential to have a secure cloud storage system that integrates a powerful cryptographic system with the efficient cloud integration in order to guarantee data privacy, integrity, secure sharing, effective access management, a good system performance, and a scalable cloud system.

1.2 Motivation

Cloud service is becoming an essential part of the modern life, where people store personal, academic, business and government information, making information security an important issue. A lack of control over data in the cloud is a growing challenge as cyberattacks, data leakage and unauthorised access to sensitive information proliferate. Hybrid cryptography, a combination of symmetric encryption and asymmetric encryption's robust key management, is an effective way of securing data stored in the cloud. The goal of this research is create a cloud storage system, which is secure, scalable and easy to use, to protect data from upload to storage, retrieval and sharing. The proposed system incorporates hybrid cryptography, access control methods, data integrity verification, and cloud services to boost trust, privacy, and security in cloud environments, while also delivering efficient and convenient data management for users.

1.3 Objectives

[1.] To develop a robust hybrid cryptography framework that combines symmetric and asymmetric encryption techniques for secure file protection.

[2.] To integrate the system with cloud platforms such as AWS, Google Cloud, and Microsoft Azure for secure file storage and management.

[3.] To design a user-friendly web and mobile interface for secure file upload, retrieval, and sharing.

[4.] To implement end-to-end encryption ensuring data confidentiality during storage and transmission.

[5.] To develop a secure access control mechanism for managing file sharing permissions and user authorization.
[6.] To incorporate data integrity verification techniques to detect unauthorized modifications and maintain file authenticity.
[7.] To ensure system scalability for handling increasing storage requirements and a growing number of users.

2. Literature Review

In Pan et al. (2015) Alongside the previously proposed subsampling method, a novel technique for image steganography incorporating compressive sensing was introduced. A novel embedding domain may serve as the recipient of a concealed message if this occurs. The attributes of Compressive Sensing (CS) were examined with regard to the potential for the original image to be compressed inside the transform domain. This was achieved by employing dimensional reduction and random projection to embed secret messages into the compressive sensing domain of a sparse image. The transmitter and receiver exchanged a confidential key to produce the measurement matrix. The stego-image would subsequently be approximately reconstructed utilizing the Total Variation (TV) minimization method. In instances when the integrity of the stego-image could not be guaranteed, a superior quality was attained by employing various transformation coefficients on the sub-images obtained from subsampling. The accuracy of the method was assessed by comparing the original and extracted messages through the Bit Correction Rate (BCR). The numerical experiments indicated that the steganography approach could offer a novel domain for data embedding while preserving a high degree of information security. [1]

Muhammad et al. (2015) Proposed a novel and significant technique for RGB photographs that utilized Grey Level Modification (GLM) and Maximum Likelihood Estimation (MLE) as its basis. Before being assigned to the grey levels of the cover pictures, supplementary secret key data was encrypted utilizing an MLEA. Upon completion, the cover picture preceding data concealment may employ an auxiliary transposition function. Malevolent users find it challenging to access original and confidential information due to the transposition, the GLM, the MLE, and the secret key, which enhance security. Both evaluations indicated that the technique improved the quality of the stego-image across many security levels, as evidenced by the comparative results. [2]

Muhammad et al. (2016) Proposed an alternative concept, the Magic LSB Substitution Method (M-LSB-SM), which was utilized for RGB images. Aside from a spatial domain Multi-Level Encryption (MLE), the proposed system utilized the achromatic component (I-plane) of the HSI colour model. Utilizing the secret key, we partitioned the I-plane into four congruent subimages and rotated each at a distinct angle. Employing MLEA, we divided all sensitive data into four separate segments. The rotational pattern-based sub-image included every sub-block employing a magic LSB substitution. Experimental results validated the effectiveness of all methods for improving stego-images and their visual quality, which consequently offered differing levels of imperceptibility and security. [3]

Muhammad et al. (2016) Proposed an enhanced adaptive LSB replacement utilizing an uncorrelated colour space to improve imperceptibility, diminish detection, and integrate human vision into the algorithm. The technique utilized an input image that was encrypted by an image scramble effect, thereby maintaining the integrity and confidentiality of the original image. The encrypted image was subsequently converted to HSV colour space and processed accordingly. The Iterative Magic Matrix Encryption Algorithm (IMMEA) generates cipher content and enhances the security of any confidential information through additional levels of protection. An adaptive LSB method was employed to integrate the encrypted data into the V plane of the HSV colour model, utilizing the LSB secret-key directed block magic methodology. Owing to the exponential rise in internet users, it will be imperative to send data in a veiled or encrypted manner. Information system security techniques are widely accessible. [4]

Muhammad et al. (2017) Offered an enhanced safe cryptographic framework for application in image steganography, utilizing colour transformation models alongside the Three-Level Encryption Algorithm (TLEA) and Least Significant Bit (LSB) substitution guided by Morton Scanning (MS). The system employed an MS-directed LSB replacement method to embed the secret data into the I-plane of the input image within the Hue-Saturation-Intensity (HSI) colour space. Enhanced security and authentication were ensured by any confidential data that had been encrypted utilizing the TLEA before its actual implementation. The quantitative and qualitative findings indicated that the enhanced processes validated the validity of visual information on social networks, and the performance was satisfactory. [5]

Muhammad et al. (2017) Introduced a further steganographic framework known as Stego Key-directed Adaptive LSB (SKA-LSB), which used multi-level cryptography as an alternative. An adaptive LSB substitution technique was employed to embed encrypted data within the host image, incorporating a stego-key encrypted via a Two-Level Encryption Algorithm (TLEA) and secret information encrypted

using a Multi-Level Encryption Algorithm (MLEA). Quantitative and qualitative results validated the framework's capability to achieve a balance between image security and quality. In comparison to other cutting-edge methodologies, their suitable and straightforward payload further validated their efficacy. [6]

Rabie et al. (2018) Devised a strategy for obscuring images that was exceptionally effective and yielded superior outcomes. In the domain of Discrete Wavelet Transform (DWT), these innovative cloaking techniques utilize a multiscale Laplacian pyramid. Enhanced capacity, however detrimental to stego quality or advanced quality, resulted from prior endeavours. The suggested technique aims to enhance the payload capacity by utilizing high-frequency bands in the Discrete Wavelet Transform of cover pictures for improved concealment. This was accomplished by concealing within the lowest tier of the Laplacian pyramid via a method in the curve-fitting adaptive area for spectral magnitude and its discrete cosine transform. In comparison to alternative competing tactics, the proposed system attained a superior capacity while enhancing visual fidelity. The testing results indicated that the proposed system might exceed all existing methods in payload capacity and image quality parameters. [7]

Sarmah (2017) Proposed two further steganographic techniques that obscured text within greyscale photographs compressed via JPEG. A 16 x 16 quantization table was utilized in lieu of the standard JPEG tables to enhance JPEG compression, which relied on a discrete cosine transform methodology. Furthermore, the study presented two additional optimization algorithms—the Multirandom Start Local Search (MRSLS) algorithm—applied to steganography based on cohort intelligence (CI) and cognitive computing (CC). The CI, originating from social learning, represents a novel optimization technique. This is tested for NP-hard unconstrained problems with significant ramifications. A novel and intriguing domain of machine learning, CC encompasses self-learning systems. This proposed study addressed NP-hard combinatorial problems by evaluating the CI, CC, and MRSLS, which were derived from the duo-swapping methodology. This study enhanced prior research by implementing the MRSLS in steganography, facilitating the testing and validation of outcomes through this and analogous methodologies. Six grayscale images were evaluated in the studies. The experiment's results revealed the stegoimages, their quality, and the capacity for inserting secret images. [8]

K. Alla et. All (2009) Proposed statistical action detection systems monitor and identify audit records that deviate from normal behaviour by analyzing the host's audit logs. The rule-based action detection system monitors the audit data for recognized actions, such as failed login attempts, root-to-user access, and remote login. Contemporary action detection systems frequently employ two methodologies, known as anomaly detection and abuse detection. A rule-based system constructed from domain knowledge regarding the signatures of previous activities is commonly employed for misuse detection. When circumstances deviate from the norm, it is perceived as an indication of malevolent intent. [9]

Kini et al. (2019) Exhibited cutting-edge computer security protocols and illustrated the superior application of concealed data compared to encrypted data. Due to their ease of tracing, LSBs are a favoured method for obfuscating information. By altering the font to integrate the Stegano key, they demonstrated the method of employing their operator's picture to obscure an additional image. The key image and the shorthand key are both encompassed by 24-bit colour picture capability. The graph analysis indicates the extent to which the steno image is obscured within the carrier image, and the findings present a comparison of the corresponding PSNR with the MSE. [10]

Swain et al. (2018), Execute a high-range steganography method utilizing a separation and replacement device. The image was segmented into overlapping 3 × 3 pixel sections. The two least significant bits in each square pixel are subjected to least significant bit substitution, while the remaining six bits undergo remainder separation. There are two tiers of installations: (i) transforming the LSB to a subordinate bit plane and (ii) turning the QVD to a superior part plane. Upon insertion along these parameters, if a square encounters a boundary drop issue, it eliminates the original mix setting and implements the converted 4-piece LSB replacement. A limitation of this study is that the proposed steganography technique eludes detection by pixel difference histogram and RS analysis tools. [11]

Singh et al. (2018), Endeavoured to amalgamate LSB with the DWT algorithm as a component of their research. Pixels in a display image might contain sensitive data embedded utilizing the least significant bit (LSB) insertion technique, which is a spatial method. Implementing this method is straightforward and uncomplicated. This article utilizes the frequency domain method as the framework for the discrete waveform transformations (DWT) technique. This transformation methodology segments the tissue and covert picture into many components, obscures the critical elements across various metrics, produces minimal embedding effects on the tissue image, and attains elevated secrecy across most ratios (PSNR). The DWT convolutional algorithm rule partitioned the image into four sub-branches, which were further

separated into the 'S', 'U', and 'V' matrix regions. Each Single Value Victimization Analysis (SVD) comprises R, G, and B components. The incorporation of color-coded segments for configuring encryption keys and confidential data enhances the attachment's reliability significantly. Comparing the performance of LSB insertion with the convergent DWT algorithm reveals efficacy in terms of both MSE and PSNR. The modified DWT technique enhances the PSNR values of encoded and encrypted images. The proposed LSB insertion strategies produce outstanding results. To enhance security, it is prudent to rewrite and re-sign the encrypted images as standard procedure. [12]

Gedkhaw et al. ((2018), Examined the methodology and techniques for clandestinely acquiring previously undisclosed information using several data formats. The researchers in this study evaluated the effectiveness of optimal disguise file size and the concealed knowledge method using a quilt image with the lowest bit rate (LSB) in the world. The results indicated that the quilt picture significantly influences file size; for example, the Display picture is 576 kb, although a 175 kb file, constituting 30% of the image, can be concealed. The display image size is 4.57 MB, allowing for the concealment of 45.95% of the image, equating to 2.10 MB. The display image possesses a resolution of 4000 x 3000 and a file size of 7.31 MB. Up to 4.29 MB, or 58.69% of the displayed image, can be concealed in this manner. The size of the patchwork image may vary; nonetheless, it remains substantial, whereas the computer file utilized to obscure the data is quite little. Future studies could consider making more algorithmic rules, such as DCT, DWT, and SPIHT, optional, as per their recommendation. [13]

Hussain et al. (2018), Championed the use of a provisioning map that is simultaneously chaotic and auxiliary to enhance the security of digital photographs. Prior to concealing the Pell sequence within the least significant bit of the hood image for enhanced security, the confidential binary data stream, derived from the iris process, is encrypted into a highly random binary sequence of uniform length. Their standard protocol entails extracting three Pell sequences from the base shell, amalgamating them into a stochastic binary sequence, and subsequently employing a provisioning map with varying initial values to obfuscate the LSB information within the basic shell image. Utilizing LSB on the most prominent picture within the Pell sequence as the primary encapsulation of the principal data stream reinstates the original values of the altered LSB bits in the master image. Consequently, the image on the inner cover remains unaltered, while constituting around 5% of the conscious information flow—an additional stream of latent knowledge. The entire process involved altering or preserving LSB bits during the execution of the cover's master image. This confidential information stream is encrypted arbitrarily. Thus, the quilt's supporting pixels generally constitute a sequential series. To safeguard the tissue image from rigorous analysis, it undergoes pre-processing before to the insertion of the LSB. Sequential pale sequences and binary random sequences possess no probability within the expansive and inscrutable core of this technology. Searching for images to identify sensitive knowledge streams is laborious, as the background Display Image is pre-processed while the original Display Image remains unaltered. [14]

George et al. (2017), Cloud computing is characterized as a pervasive and expansive network of multiple interconnected systems that use at least one resource of record driven by demand-based utilization. Cloud assets are not invariably beneficial, even when customers and cloud services exhibit more compatibility. Safety is a paramount concern among them. Issues regarding data privacy and security, universal accessibility, disaster recovery, and corporate progress are essential to the evolution of cloud-based data storage. Ensuring customer data security can be achieved using traditional methods; but, as cloud users become more concerned about their data, its safeguarding emerges as a critical issue that necessitates thorough scrutiny. The RSA algorithm exemplifies an asymmetric key algorithm, utilizing two distinct keys for encryption and decoding. Enhancing the encryption cycle is achievable by altering key dimensions. This renders data collection a challenging endeavour for the coder. The duration of encryption and decryption escalates with the rise in the number of keys. The dividing of the text into smaller segments enhances the efficiency of the move algorithm for encryption and decryption, while concurrently fortifying the process with larger key sizes. Eastern Force prepares to facilitate seamless cloud data storage for consumers. [15]

3. Methodology

3.1 Introduction

This chapter delineates the methods employed for the design and implementation of a secure cloud-based file storage system utilizing a hybrid cryptography approach. The objective is to guarantee data confidentiality and secure access via a mix of contemporary symmetric encryption methods and stringent key management practices. The methodology encompasses several processes, including file preprocessing, chunking, encryption, secure key generation, decryption, and restoration. Each phase has been methodically crafted and included into a Flask-

based web application to facilitate user interaction and testing.

3.2 System Architecture

The overall architecture of the system consists of six major modules:

- **File Upload Module**
- **File Divider Module**
- **Hybrid Encryption Module**
- **Key Generation and Management**
- **Decryption and Restoration Module**
- **User Interface (Web-based Interaction)**

3.3 Flowchart of the System Process

The following flowchart illustrates the logical flow of data and operations from file upload to final restoration:

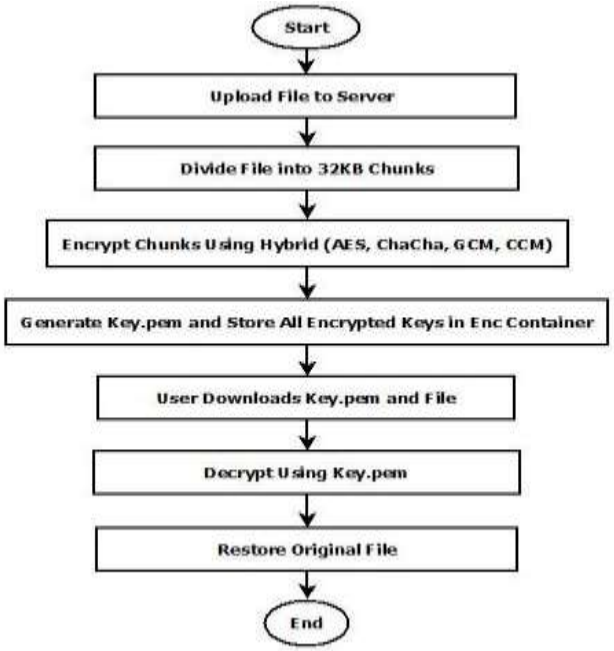


Figure: 3.1 Overall System Flowchart

3.4 Modules Description

3.4.1 File Upload Module

Users begin by uploading a file through the web interface. The file is stored temporarily in the uploads/ directory. The system checks the file format and processes it for further operations.

3.4.2 File Divider Module

Once uploaded, the file is split into fixed-size chunks (32 KB) using the divider.py script. The number of chunks and metadata are recorded in raw_data/meta_data.txt. This makes the encryption scalable and suitable for large files.

3.4.3 Hybrid Encryption Module

Each chunk is encrypted sequentially using four different algorithms in rotation:

- **AES with MultiFernet key rotation**
- **ChaCha20-Poly1305**
- **AES-GCM**
- **AES-CCM**

This hybridization ensures that no single encryption method is responsible for all data, improving resistance against cryptanalysis.

Encrypted files are saved in the encrypted/ directory.

3.4.4 Key Generation and Management

A Fernet key (key_1) is used to encrypt all the session keys and nonces used during encryption. This combined encrypted keyset is saved as store_in_me.enc.

The key_1 is then stored in a downloadable .pem file called Main_Key.pem. This key is critical and must be securely transferred to the receiver for successful decryption.

3.4.5 Decryption and Restoration Module

To recover the original file:

- The user uploads the Main_Key.pem.
- The system decrypts the secret key set from store_in_me.enc.
- It reverses the hybrid encryption on each file chunk.
- Finally, the decrypted chunks are reassembled to form the original file.

3.4.6 User Interface and Interaction

A simple Flask web interface allows:

- Uploading files
- Downloading encryption key
- Uploading the decryption key
- Downloading the restored file

User feedback and success messages ensure a smooth experience. Key operations are confirmed with visual indicators and download buttons, as shown in the screenshots in the results chapter.

3.5 Summary

This Section outlined the detailed methods employed for the design and implementation of a secure file storage system utilizing hybrid cryptography. The suggested methodology focuses on integrating various symmetric encryption algorithms, modular system design, and an interactive web interface to ensure robust security, reliability, and user accessibility.

The process commences with a File Upload Module, wherein the user selects a file for encryption via an intuitive online interface. This file is briefly retained on the server for processing. The system guarantees appropriate management of the submitted data and verifies its structure prior to further processing.

In the subsequent phase, the File Divider Module is essential for facilitating effective encryption by segmenting the uploaded file into fixed-size portions of 32KB. This chunking method enables parallelizable and scalable encryption for larger files. It introduces an extra layer of complexity by allowing each data element to be encrypted independently with distinct algorithms.

Subsequently, the Hybrid Encryption Module encrypts the file segments utilizing a cyclic process that

incorporates four robust encryption methods: AES (MultiFernet), ChaCha20-Poly1305, AES-GCM, and AES-CCM. Each algorithm is utilized in a sequential rotation throughout various segments, enhancing security by circumventing the constraints of dependence on a singular cryptographic technique. The utilization of distinct session keys and nonces enhances encryption by guaranteeing randomization and mitigating pattern identification.

The Key Generation and Management Module oversees the production of all cryptographic keys utilized in encryption. All session keys and initialization vectors (nonces) are concatenated and subsequently encrypted with a Fernet key. The key is subsequently stored in a .pem file (Main_Key.pem) and provided to the user for download. This method guarantees that only an individual possessing the original key file can decode and recover the encrypted data, hence maintaining stringent access control protocols.

The Decryption and Restoration Module is tasked with reversing the encryption process. Upon receipt of the uploaded Main_Key.pem file, the system decrypts the master secret, extracts all individual keys, and employs the appropriate decryption technique for each encrypted segment. Ultimately, the system reassembles the original file from the decrypted segments, enabling the user to download it securely.

The system is facilitated by a clear, user-friendly interface created with Flask. This interface facilitates seamless navigation for file uploads, key downloads, decryption key uploads, and retrieval of the recovered file. Success announcements, error alerts, and guided directions improve the user experience and reduce the likelihood of errors during essential activities.

This methodology chapter delineates a structured, modular, and safe framework for file storage and retrieval employing hybrid cryptography. It tackles fundamental issues in data confidentiality, access management, and usability. The design decisions implemented in this system—including file chunking, rotating encryption, one-time key download, and user-centric design—collectively establish a robust foundation for creating a safe, scalable, and deployable solution for cloud-based file security.

4. Results

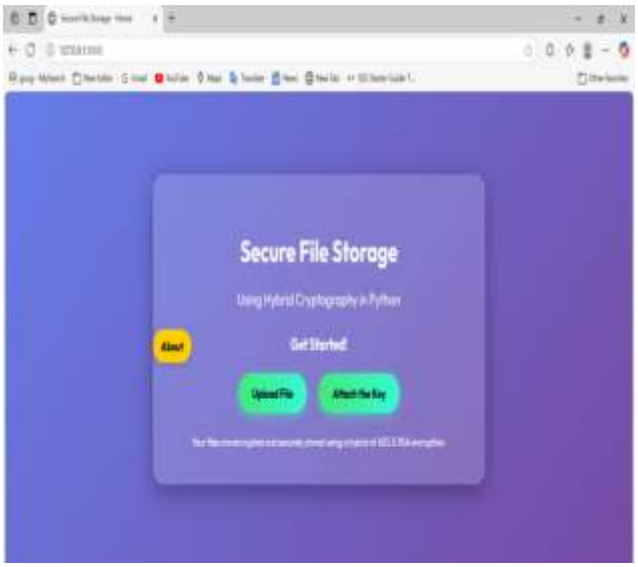


Figure: 4.1 Home Page – Secure File Storage Landing Interface

The graphic above illustrates the homepage of the secure file storage system. The interface is aesthetically pleasing, showcasing a vivid gradient background and a centralized card arrangement. The program title “Secure File Storage” is prominently featured at the centre, along by the subtitle “Using Hybrid Cryptography in Python,” which explicitly conveys the project's aim. The interface offers three navigation choices: “About,” “Upload File,” and “Attach the Key.” These buttons guide users to several phases of the system—delivering information, commencing file encryption, and initiating the decryption process, respectively. The homepage functions as the principal access point for users and establishes the ambiance for a streamlined, user-centric experience.

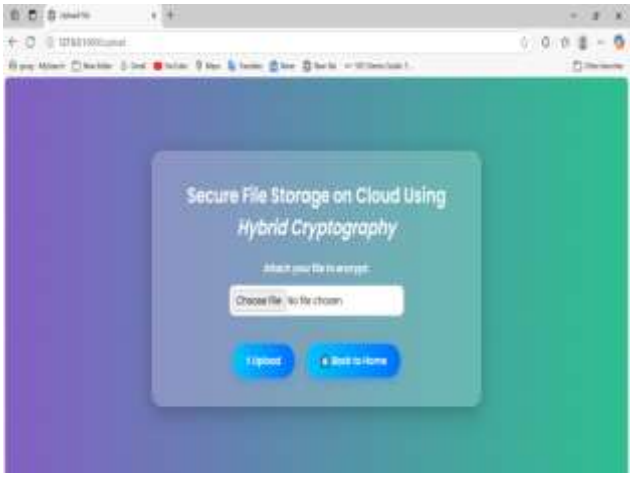


Figure: 4.2 Upload Interface – File Selection Window

This screen is displayed when the user selects the “Upload File” button on the home page. It is intended to enable users to upload a file for encryption. The

interface maintains the identical vibrant gradient background and visual allure. The title "Secure File Storage on Cloud Using Hybrid Cryptography" emphasizes the fundamental principle of the project. Below, there is a file input box accompanied by two buttons: "↑ Upload" to initiate encryption and "Back to Home" to abort the procedure. No file has been selected at this juncture, and the user is invited to select a file from their local system. The design is uncomplicated and user-friendly, facilitating effortless file selection for encryption.

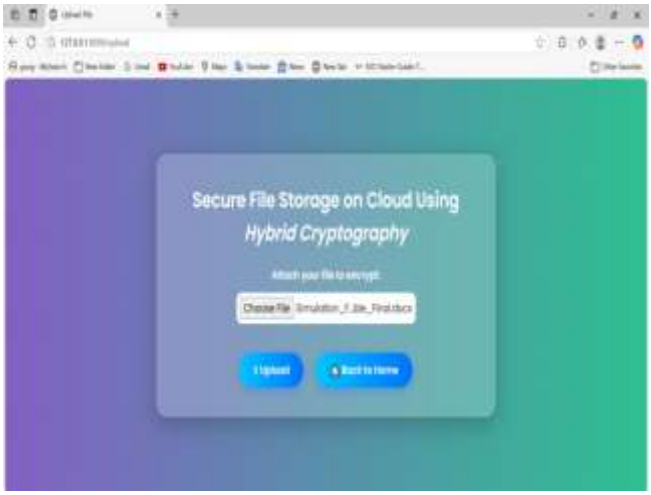


Figure: 4.3 Upload Interface – File Attached and Ready to Encrypt

This screenshot depicts the time subsequent to the user's selection of a file for encryption. The document titled Simulation_F...ble_Final.docx has been selected, as indicated in the file input box. The page structure is identical to the preceding screen, including a uniform design and button alternatives. Clicking the "↑ Upload" button transmits the file to the backend for processing. The file will be partitioned, encrypted with multiple cryptographic methods (including AES, ChaCha20, AES-GCM, and AES-CCM), and subsequently saved safely. This screen enables users to verify their file selection prior to executing irreversible encryption.

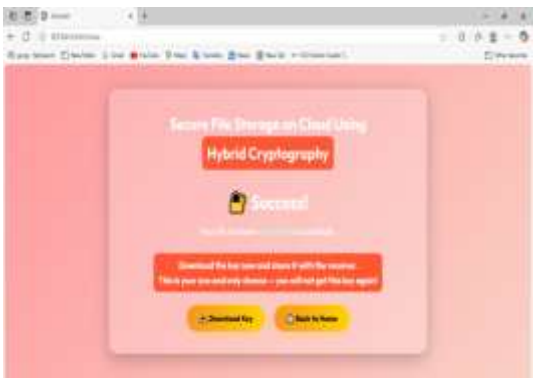


Figure: 4.4 Success Page – Encryption Complete and Key Download

The figure above illustrates the interface displayed following a successful file encryption procedure. The file has been securely encrypted using hybrid cryptography and is currently stored in the system. The caption "Success!" is prominently shown, followed by a lock icon to emphasize the theme of security. Below this, a crucial command is presented in red: "Download the key immediately and disseminate it to the recipient." This is your sole opportunity - you will not receive this key again! This underscores the importance of the encryption key (Main_Key.pem) for subsequent file decryption. Two buttons are available: "Download Key" for obtaining the encryption key and "Back to Home" for returning to the main interface. This page concludes the encryption process and emphasizes the necessity of safeguarding the key.

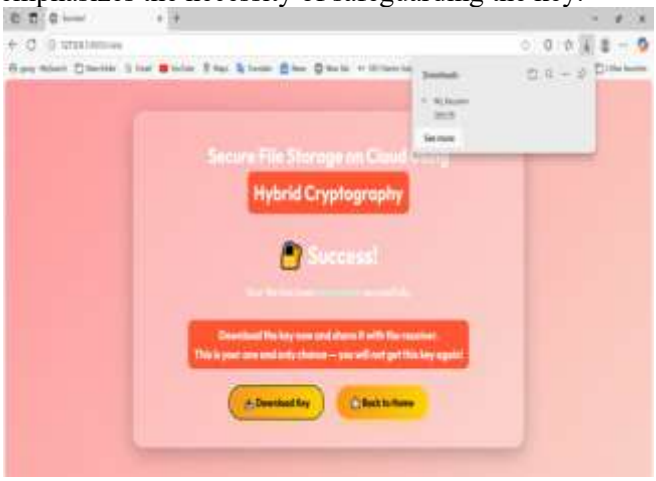


Figure: 4.5 Downloaded Generated Key

The aforementioned figure illustrates The online program verifies that the file has been encrypted with a hybrid cryptographic method, integrating several contemporary encryption algorithms such as AES, ChaCha20-Poly1305, AES-GCM, and AES-CCM. Upon clicking the download button, the key will be downloaded.

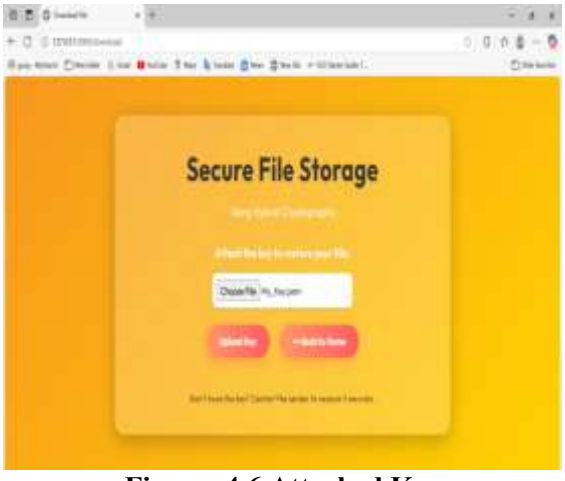
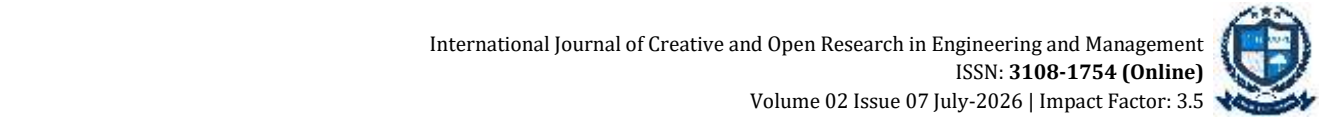


Figure: 4.6 Attached Key



This figure illustrates that the encryption key has been successfully generated and securely affixed to the system as a downloadable file titled My_Key.pem. This key is an essential element of the hybrid cryptographic system utilized in the application. It contains the confidential data necessary to decrypt the segmented encrypted files, which were previously partitioned and secured with a series of sophisticated encryption methods (AES with key rotation, ChaCha20-Poly1305, AES-GCM, and AES-CCM).

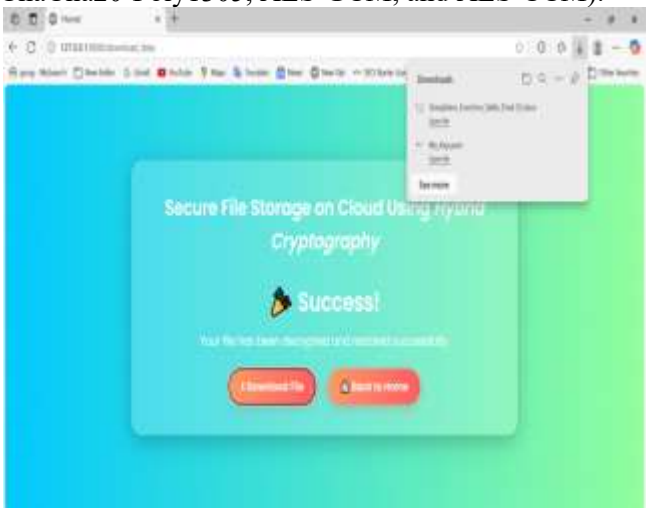


Figure: 4.7 Download Decrypted File

This screen demonstrates that the complete hybrid cryptographic framework—from file upload, segmentation, encryption, secure key transmission, to decryption and file restoration—has operated as designed. The solution effectively safeguards sensitive data during storage and facilitates permitted recovery with appropriate credentials, underscoring both the robustness of security and the efficiency of usability in your research prototype.

Discussion

The main aim of this research was to develop and execute a safe, efficient, and user-friendly system for cloud-based information storage via a hybrid cryptography method. The suggested approach effectively integrates many contemporary encryption techniques to guarantee the security, integrity, and authenticity of user data, while also enhancing usability via a cohesive online interface.

The encryption process entails segmenting the uploaded file into manageable portions and subsequently applying several cryptographic algorithms in succession to each segment. The system employs AES with key rotation (via MultiFernet), ChaCha20-Poly1305, AES-GCM, and AES-CCM. This diversity enhances the complexity for any opponent engaging in cryptanalysis, as the ciphertext is safeguarded by many algorithms and keys. The encryption keys and initialization vectors (nonces) are safely produced and subsequently encrypted with an additional Fernet key, which is sent to the user as a .pem file (e.g., My_Key.pem). The system mandates a one-time key download to reduce exposure and enhance secure key sharing protocols.

The initial screenshot demonstrates that the successful encryption confirmation screen verifies to the user that their data is secured and securely stored. The system emphasizes the necessity of downloading the key promptly, as it will not be accessible thereafter. This highlights users' responsibilities in key management, underscoring zero trust and non-redundancy, which are fundamental to contemporary cryptographic standards.

In the subsequent phase, the decryption module activates upon the user's submission of the appropriate .pem file. The application decrypts the encrypted keys, reconstructs the original keys and nonces, and reverses the hybrid encryption process to retrieve the original file. The second snapshot verifies the successful decryption and restoration of the original file, which the user can then download. The outcome not only validates the algorithm's accuracy but also demonstrates the system's comprehensive reliability, encompassing encryption, decryption, and recovery processes.

This modular and flexible architecture offers a robust basis for the implementation of secure cloud services. The system's notable accomplishment is its resilience to brute-force and single-point-of-failure assaults, attributed to its hybrid and multi-algorithm encryption strategy. Furthermore, the method preserves simplicity in user engagement, guaranteeing that even non-expert users may safely encrypt and decode information with low exertion.

Nonetheless, despite the security layer's robustness, the system's dependence on secure key management and distribution continues to pose a significant difficulty. In practical implementation, integrating blockchain-based key management or secure multi-party computation could significantly augment the system's robustness and decentralization.

5. Conclusion and Future Scope

Conclusion

This study offers a comprehensive and effective solution for safe cloud file storage with a hybrid cryptographic method. The suggested system optimizes data secrecy and offers multilayer security against diverse cryptographic threats by integrating many modern encryption algorithms, including AES (with MultiFernet key rotation), ChaCha20-Poly1305, AES-GCM, and AES-CCM. The novel application of cyclic encryption on file segments enhances resistance to cryptanalysis by averting consistent encryption throughout the whole file.

The project illustrates the effective coexistence of security and usability. The intuitive web interface facilitates seamless file uploads, automated encryption, safe key creation, and straightforward decryption via a guided experience. The solution guarantees comprehensive file protection, from the first upload to the ultimate download of the decrypted content.

The one-time downloading key mechanism enhances secure key management by reducing the possibility of unauthorized access. The positive outcomes, as demonstrated in the interface pictures, confirm the system's capacity to manage secure file transactions consistently and effectively.

This study establishes a robust framework for secure cloud storage systems and can function as a prototype for implementation in real-world contexts where data privacy is paramount, including healthcare, financial, legal, and government sectors. This technique adheres to contemporary cryptographic standards and establishes a foundation for future advancements, such as decentralized key sharing, biometric identification, and blockchain integration for immutable auditing.

Future Scope

Future prospects include various prospective avenues to augment the capabilities and resilience of the existing system. A crucial aspect is the enhancement of mobile and cross-platform compatibility, which would greatly broaden the application's usability. By facilitating secure encryption, uploading, and retrieval of files from smartphones, tablets, and various operating systems, the system may accommodate a wider and more diversified user demographic.

A significant improvement is found in the incorporation of compression and optimization techniques. Utilizing data compression algorithms prior to encryption may diminish the overall file size, hence enhancing performance, particularly when managing substantial multimedia or document files in bandwidth-constrained contexts. This would enhance the system's efficiency and accessibility for users with limited network resources.

As the cybersecurity landscape evolves, it is imperative to prepare for evolving attacks. As quantum computing progresses, conventional cryptography techniques may prove inadequate. Consequently, subsequent iterations of the system ought to integrate quantum-resistant cryptographic methods, particularly post-quantum cryptography (PQC), to guarantee enduring security and robustness against quantum assaults.

Ultimately, the implementation of audit logs and usage monitoring would introduce a significant layer of transparency and oversight. By monitoring user activity, file access patterns, and key utilization, the system can facilitate adherence to security policies, assist in troubleshooting, and identify potential misuse or illegal actions. These improvements collectively facilitate a more secure, scalable, and user-focused solution for cloud-based data security.

References

[1.] Pan, JS, Li, W, Yang, CS & Yan, LJ 2015, ‘Image steganography based on subsampling and compressive sensing’. Multimedia Tools and Applications, vol. 74, no. 21, pp. 9191-9205.

[2.] Muhammad, K, Ahmad, J, Farman, H, Jan, Z, Sajjad, M & Baik, SW 2015, ‘A Secure Method for Color Image Steganography using GrayLevel Modification and Multi-level Encryption’. TIIS, vol. 9, no. 5, pp. 1938-1962.

[3.] Muhammad, K, Sajjad, M, Mehmood, I, Rho, S & Baik, SW 2016, ‘A novel magic LSB substitution method (M-LSB-SM) using multi-level encryption and achromatic component of an image’. Multimedia Tools and Applications, vol. 75, no. 22, pp. 14867-14893.

[4.] Muhammad, K, Sajjad, M, Mehmood, I, Rho, S & Baik, SW 2016, ‘Image steganography using uncorrelated color space and its application for security of visual contents in online social networks’. Future Generation Computer Systems, vol. 86, pp. 951-960.

[5.] Muhammad, K, Ahmad, J, Rho, S & Baik, SW 2017, ‘Image steganography for authenticity of visual contents in social networks’. Multimedia Tools and Applications, vol. 76, no. 18, pp. 18985-19004.

[6.] Muhammad, K, Ahmad, J, Rehman, NU, Jan, Z & Sajjad, M 2017, CISSKA-LSB: color image steganography using stego key-directed adaptive LSB substitution method. Multimedia Tools and Applications, vol. 76, no. 6, pp. 8597-8626.

[7.] Rabie, T, Baziyad, M & Kamel, I 2018, Enhanced high capacity image steganography using discrete wavelet transform and the Laplacian pyramid. Multimedia Tools and Applications, vol. 77, no. 18, pp. 23673-23698.

[8.] Sarmah, DK & Kulkarni, AJ 2017, ‘Image steganography capacity improvement using cohort intelligence and modified multi-random start local search methods’. Arabian Journal for Science and Engineering, pp. 1-24.

[9.] K. Alla and R. S. R. Prasad, “An Evolution of Hindi Text Steganography,” in 2009 Sixth International Conference on Information Technology: New Generations, Apr. 2009, pp. 1577–1578. doi: 10.1109/ITNG.2009.41.

[10.]S. D. Hu and K. Tak U., “A Novel Video Steganography Based on Nonuniform Rectangular Partition,” in 2011 14th IEEE International Conference on Computational Science and Engineering, Aug. 2011, pp. 57–61. doi: 10.1109/CSE.2011.24.

[11.] Swain Gandharba (2018), "Very High Capacity Image Steganography Technique Using Quotient Value Differencing and LSB Substitution ", Springer Arabian Journal for Science and Engineering, pp- 1-8.
[12.] Singh Akanksha, Chauhan Monika and Shukla Shilpi (2018), "Comparison of LSB and Proposed Modified DWT Algorithm for Image Steganography ", IEEE International Conference on Advances in Computing, Communication Control and Networking (ICACCCN2018), PP-889-893.
[13.] GedkhawEakbodin, SoodtoetongNantinee and Ketcham Mahasak (2018), "The Performance of Display-Image Steganography for Hidden Information within Image File using Least Significant bit algorithm", IEEE the 18th International Symposium on Communications and Information Technologies (ISCIT 2018), pp- 504-508, 2018.
[14.] Anwar Md. Hussain and Bora Popi (2018), "A Highly Secure Digital Image Steganography Technique Using Chaotic Logistic Map and Support Image", Proceedings of IEEE International Conference on Information Communication and Signal Processing (ICSP 2018), pp-69-73
[15.] George Dr. D.I., Leena H. M. and Amalarethinam (2017),” Enhanced RSA Algorithm with varying Key Sizes for Data Security in Cloud”, IEEE World Congress on Computing and Communication Technologies (WCCCT), pp172-175.
[16.]Fei Gao, Su-Juan Qin, Fen-ZhuoGuo & Qiao-Yan Wen 2010, ‘DenseCoding Attack on Three-Party Quantum Key Distribution Protocols’, IEEE Journal of Quantum Electronics, vol. X, no. X, pp. 1-6.
[17.]Ananda Rao, G, Srinivas, Y, VijayaSekhar, J & Pavan Kumar, CH 2011, ‘Three Party Authentication Key Distributed Protocols Using Implicit and Explicit Quantum Cryptography’, Indian Journal of Computer Science and Engineering, vol. 2, no. 2, pp. 143-145.
[18.]Sathi Reddy, K & Raja Kumar Medapati 2011, ‘Three Party Quantum Authenticated Key Distribution Protocol Using Superposition States’, Int. J. Comp. Tech. Appl., vol. 2, no.5, pp.1589-1594.
[19.]Wang Jian, Zhang Quan & Tang Chao-Jing 2007, ‘Multiparty Quantum Secret Sharing of Secure Direct Communication Using Teleportation’.
[20.]Abdesslem Layeb 2011, ‘A novel quantum inspired cuckoo search for Knapsack problems’, International Journal of Bio-Inspired Computation, vol. x, no. x.
[21.] Khumani, Priya Roy, Swapnil “SECURE FILE STORAGE ON CLOUD USING HYBRID CRYPTOGRAPHY” International Journal of Multidisciplinary Research (IJMR) 2021.
[22.] Aditya SadanandGhadi “Secure File Storage Using Hybrid Cryptography” International Journal of Innovative Science and Research Technology2020.