Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 47 No. 02 (2026)

# Secure File Transfer Using Hybrid Cryptography-Based Web App

**Mrs. Khushi Gharat*, Mr. Ajay Sirsat**

M.Tech Student, **C**omputer, St. John college of Engineering & Management, Palghar, India

Professor, **C**omputer, St. John college of Engineering & Management, Palghar, India

**Abstract:-** Secure file management has become a key concern in the current digital landscape, where sensitive information is exchanged and stored regularly for personal, academic and sprofessional activities. In addition to holding such sensitive information as business contracts, health records, legal documents and financial information, the potential for confidentiality of these types of information is increasing due to cyber threats from unauthorized access, phishing attacks, ransomware and insider leaks. The above-mentioned data risks demand effective systems to protect files from being tampered with, stolen or misused, while at the same time allowing users to continue access and control of the files they create or manage.

Typical file storage solutions have restrictions on how securely they can provide file storage and may use proprietary methods of encryption/security that lack transparency and strong encryption/protective controls, thus exposing confidential information to potential data breaches. The mentioned problems can be solved through cryptography by using systematic methods to convert human-readable information into unreadable but secure formats to ensure the confidentiality, integrity, and authenticity of the information stored.

**Keywords**: Advanced Encryption Standard (AES), Rivest-Shamir-Adleman (RSA)

## 1. Introduction

With today's digital data exchange as a common practice and the foundation of all personal, academic and professional activities, safeguarding sensitive files is now a fundamental requirement. Whether it's a business contract, a health record, or any other confidential document, organizations and individuals alike must continuously monitor their file security to ensure no one can gain unauthorized access, or manipulate confidential data. Cybersecurity threats of all types such as unauthorized access, phishing, ransomware and insider leaks contribute to the increasing urgency of having developed and established systems for securely managing files.

While traditional file storage solutions are a convenient method of storing files, often they're lacking built-in mechanisms to provide strong encryption capabilities, or rely on proprietary security practices that aren't transparent to end users. As such, there has been growing demand for applications that allow individuals and organizations complete control of their data's security. This is where cryptography plays an important role in satisfying that demand by providing standardized ways to convert information into an unreadable format while maintaining the following qualities:confidentiality, integrity, and authenticity.

Of the various methods used to protect data with cryptography, symmetric encryption (i.e., AES) offers the speed and efficiency necessary to process large files. However, symmetric encryption has a key distribution issue because the same secret key must be securely exchanged between both sender and recipient. In contrast, asymmetric encryption (i.e., RSA), provides a better solution to the key distribution problem by using different keys for different types of encryption. For example, the sender generates a public/private keypair during asymmetric encryption; the public key is used to encrypt the message, the private key is retained only by the owner of the private key, and the recipient uses it to decrypt the message.

## 2. Methods

The methodology of the Secure File Transfer System proposed in this document incorporates the use of hybrid cryptography to create an online solution for file transfers that incorporates both confidentiality and secure key management. This is accomplished through the use of symmetric cryptography to provide speed and efficiency when transferring files, while still utilizing the asymmetric characteristics of asymmetric cryptography to protect the keys used in the system (e.g., managing the use and limits of each user's cryptographic keys). The Advanced Encryption Standard (AES) was implemented to encrypt the contents of documents transferred over a secure connection using AES with a key length of 256 bits (i.e., symmetric encryption), while the Rivest-Shamir-Adleman (RSA) algorithm was implemented as a method to encrypt and ensure that the AES key used to encrypt the document is not used inappropriately. The frontend application of the Secure File Transfer System was developed using the Flask framework as a web application utilizing Python programming language, and with the implementation of Tiny DB used as the database used to store each user's user profile information, as well as metadata concerning the files uploaded by each user.

1. Total System Design and Implementation

The total system architecture is designed around a modular system design, with each functional module performing the functions of the secure file transfer system (e.g., Register User, Authenticate User, Upload File, Store File, Encrypt/Decrypt File, Download File). In this instance, the modular design utilizes Flask to create an application that processes Hypertext Transfer Protocol (HTTP) requests, establishes and manages a session for each user interacting with the system using an HTTP session, and to define separate routes for each of the five above-mentioned modules. The modular design enhances the maintainability, extensibility, and testability of the secure file transfer application.

For each registered user using the Secure File Transfer Application, a unique system workspace is created on the server. A workspace is a specified directory on the secure file transfer application server that contains the user encrypted files, the user encrypted keys, and other associated cryptographic data that has been created/generated by or on behalf of the user.

**Results & Discussion**

Users are able to create accounts with a unique username and password to gain access to their dashboards. Passwords are stored using a secure hashing algorithm to protect the user's credentials from being stolen by someone else. If a registered user returns, they will log in with their username and password in order to see their dashboard.

After the user is authenticated (logged in), they will be able to upload files. When the user has successfully uploaded their file, the system will also create for that user a randomly generated AES and AES Key to encrypt the uploaded file. At that point, the AES Key will be encrypted using the system created RSA Public Key. The encrypted file and encrypted AES Key will be saved to the directory of the 16 user's profile and each user will get a confirmation that their file was encrypted successfully. In addition to the confirmation they will be able to download the encrypted file, the encrypted AES Key, and the RSA generated Public & Private Key.

There is a directory called storage, where each user's encrypted files, each user's RSA keys, and each user's AES Key files will be stored. This allows for each user's storage directory to be isolated from the storage directory of another user and prevents any unauthorized user from accessing the directory. It is required for the user to supply the encrypted file, the encrypted AES Key, and the user's private RSA Key in order to decrypt an encrypted file. The private RSA Key for the user will then be used to decrypt the AES Key, which will then allow the user to decrypt the encrypted file. As a result, the user will be able to obtain a copy of the original file in its un-encrypted form through this process.

The following security measures have been put into place to further protect the user's encrypted files from unauthorized access. Additionally, all methods that can use the logged-in user's session ID must be protected with

a secure naming convention to help ensure that a user can only access their own files. Users will only have access to their own files, and other uses will not have access to the encrypted files in the same user's storage directory.

3.Literature Review

According to [2] This file will now be encrypted using Encryption Algorithm and a key for the same will be generated using RSA Key Technique. The encryption algorithm used is AES (Advanced Encryption Standard). This file is then stored to cloud server. A link is then generated for sharing the file. The link can be used to share the file to other users or also to decrypt the file. When the link is accessed through the application it prompts for the key. The key is then authenticated and then the file is downloaded and decrypted into users' local storage. [4] In this development "To decrypt the file, the authorized user needs to download the file and use the master symmetric key to decrypt the keys, and using the keys they can decrypt and obtain the divided parts of the file that is later merged into the original file through our web application.".[5]

"The proposed software product is liable to meet the required security needs of data center of cloud. Blowfish used for the encryption of file slices takes minimum time and has maximum throughput for idea of splitting and merging adds on to meet the principle of data security. The hybrid approach when deployed in cloud environment makes the remote server more secure and thus, helps the cloud providers to fetch more trust of their users."[8] The research process aims to detect cloud storage security using hybrid cryptography. In this scheme, there is the use of symmetric key cryptography and stenography techniques. This paper's content is highly focused on the security of files in the cloud.[9] In this paper, LSB Steganography technique algorithm are used. Create a various models and analysis are made. Then data or file are transferred using Hybrid Cryptography. The AES algorithm has maximum block size of 256 bits whereas Key size is unlimited. The AES design is based on a substitution-permutation network (SPN) and does not use the Data Encryption Standard (DES) Feistel network, thus making it stronger and faster than Triple-DES.

4.Figure

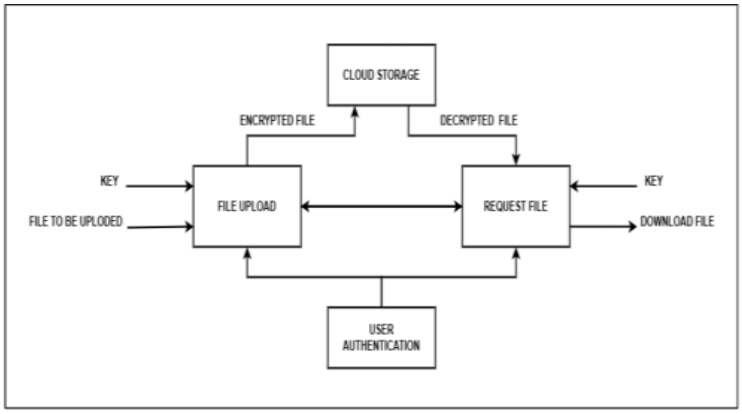


**Fig.1**

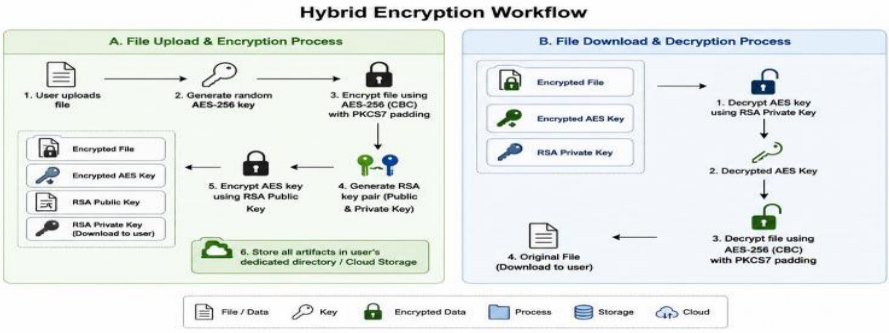


**Fig.2**

**Refrences**

[1] Jian-Foo Lai1, Swee-Huay Heng2* "Secure File Storage On Cloud Using Hybrid Cryptography" Vol. 1 No. 2 (September 2022)

[2] Abdeldime M.S. Abdelgader, Lenan Wu, Mohamed Y. E. Simik and Asia Abdelmutalab"Design of a Secure File transfer System UsingHybrid Encryption Techniques" Vol II WCECS 2015, October 21-23, 2015, San Francisco, USA

[3] Prathamesh Jagtap[1], Harshal Patil[2] "Hybrid Cryptography Algorithm Based Secured Storage Android App" Volume: 09 Issue:06 | June 2022

[4] Dr. Chandrakala B M, Aaryan Singh Rajput, Ananya M, Abhijeet Kumar, Aditya Sinha "Secure File Storage Using Hybrid Cryptography"©2024 IJCRT | Volume 12, Issue 4 April 2024

[5] Prof. Sagar Dhanake, Anushka Shivraj Kalandikar, Sara Santosh Kharchane, Apurva Sunil Salunkhe, Sanjana Sunil Jadhav"Secure File Storage on Cloud using Hybrid Cryptography"Volume 3, Issue 16, May 2023

[6] Ashwin Satheesh Kumar[1], Anfah K.[1], Hariharan T.[1], Rosna Parveen[1], Sizan Mahmud[1] and Dr. Sonal Sharma[1,2] "SECURE FILE STORAGE ON CLOUD USING HYBRID CRYPTOGRAPHY"*ISSN: 2320-540 Int. J. Adv. Res. 11(04), 01-05*

[7] Sunil Kumar and Dilip Kumar"Securing of Cloud Storage Data Using Hybrid AES-ECC Cryptographic Approach" 15 November 2022

[8] Aditya SadanandGhadi "Secure File Storage Using Hybrid Cryptography" Volume 5 Issue 12, December–2020

[9] Annsheela[1], S Habeeb Mohamed SathaAmina[2]"Establishing a Secure File Transfer Using Hybrid Cryptography and LSB Stenographic Techniques"

[10] Arun Kumar Reddy, [2]B. Sudeepthi, [3]M. Bipin Chandra, [4]N. Charan Raju "SECURE FILE TRANSFER SYSTEM" Volume 13, Issue 11 November 2025

[11] R. Kaur and J. Kaur, "Cloud computing security issues and its solution: A review," 2015 Second International Conference on Computing for Sustainable Global Development (INDIACom), 2015, pp. 1198-1200.

[12] A. J. Nathan and A. Scobell, "2020 Data Breach Investigations Report," Verizon, 2020.

[13] D. P. Timothy and A. K. Santra, "Ahybrid cryptography algorithm for cloud computing security," 2017 International Conference on Microelectronic Devices, Circuits and Systems (ICMDCS), 2017, pp. 1-5.

[14] P. Loshin, "Selected FAQs on using GnuPG," Simple Steps to Data Encryption: A Practical Guide to Secure Computing, pp. 11-21, 2013.

[15] V. S. Mahalle and A. K. Shahade, "Enhancing the data security in cloud by implementing hybrid (RSA & AES) encryption algorithm," 2014 International Conference on Power, Automation and Communication (INPAC), 2014, pp. 146-149.

[16] E. Jintcharadze and M. Iavich, "Hybrid implementation of Two fish, AES, ElGamal and RSA Cryptosystems," 2020 IEEE East-West Design and Test Symposium (EWDTS), 2020, pp. 1-5.

[17] A. Orobosade, T. Aderonke, A. Boniface and A. Gabriel, "Cloud application security using hybrid encryption," Communications on Applied Electronics, vol. 7, no. 33, pp.25-31, 2020.

[18] J. Daemen and V. Rijmen, "The block cipher Rijndael," Smart Card Research and Applications (CARDIS), 1998, Lecture Notes in Computer Science, vol. 1820, pp. 277-284.

[19] J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti and E. Roback, "Report on the development of the Advanced Encryption Standard (AES)," Journal of Research of the National Institute of Standards and Technology, vol. 106, no.3, pp. 511-577, 2001.

[20] T. ElGamal,"Apublic key cryptosystem and a signature scheme based on discrete logarithms," IEEE Transactions on Information Theory, vol. 31, no. 4, pp. 469-472, 1985.

[21] W. Penard and T. van Werkhoven, "On the secure hash algorithm family," Cryptography in Context, pp. 1-17, 2008.

[22] P. P. Pittalia,"Acomparative study of hash algorithms in cryptography," International Journal of Computer Science and Mobile Computing, vol. 8, no. 6, pp. 147-152, 2019.

[23] M. Stevens, E. Bursztein, P. Karpman, A. Albertini and Y. Markov, "The first collision for full SHA-1," Advances in Cryptology, (CRYPTO), 2017, Lecture Notes in Computer Science, vol. 10401, pp 570–596.

[24] C. Severance, "Inventing PHP: Rasmus lerdorf." Computer, vol. 45, no. 11, pp. 6-7, 2012.

[25] L. Moroney, "The firebase realtime database," The Definitive Guide to Firebase, pp. 51-71, 2017.