

Secure File Storage in Cloud Computing Using Hybrid Cryptography

J Component Project

Gokul Gopakumar 19BCI0187

Anirudh Jaykumar 19BCI0257

Atla Venkat Suhas 19BCI0205

Submitted to
Prof. AjuD

B.Tech. Computer Science and Engineering



School of Computer Science and Engineering

Vellore Institute of Technology, Vellore

April, 2022

Abstract

We exchange data among numerous clients, servers, and individuals in Cloud Computing. As a result, the security of information stored in the cloud cannot be guaranteed allowing an attacker to easily access and destroy the first type of data. Both files and software are not totally confined on the user's machine in Cloud computing. Because both the user's application and program are stored on the provider's facilities, file security problems emerge.

This difficulty can be solved by the cloud service encrypting the data with an encryption technique. In this project we explore a security model in order to provide a practical solution to the basic issue of cloud security. A multithread hybrid encryption is used in this architecture, where the file uploaded is split into sections and encrypted using different encryption algorithms like as AES, Blowfish, Triple DES, IDEA, and Fernet for safe communication between clients and servers.

File splitting and merging mechanisms, as well as multi-threading to encrypt files simultaneously, are included in the suggested technique. The file type encrypted in the project will be a text file.

Introduction

With the advent of cloud computing, there has been a surge in the number of organizations and start-ups producing cloud-based apps and products. Storage is one of the most essential services given by cloud computing firms; services such as S3, Glacier, and others are widely utilized and easily expandable. We want to offer such a system through our project, which functions as an efficient storage system for our files, encrypting them into multiple sections and combining the decrypted files as needed. This would give us with optimum file security while also allowing us to learn more about cloud security and encryption technologies.

In Cloud computing, both files and software are not fully contained on the user's computer. File security concerns arise because both user's application and program are residing in provider premises. The cloud provider can solve this problem by encrypting the files by using encryption algorithm. The use of a single algorithm is insufficient to provide high levels of security. If we employ a single symmetric key encryption technique, we will run into a security issue since this algorithm uses a single key to encode and decode data. As a result, when sharing a key in a multiuser scenario, key transmission issues arise. Although public key cryptography techniques provide great security, they need the least amount of time to encode and decode data.

To address the aforementioned vulnerabilities, we've implemented a new security method.

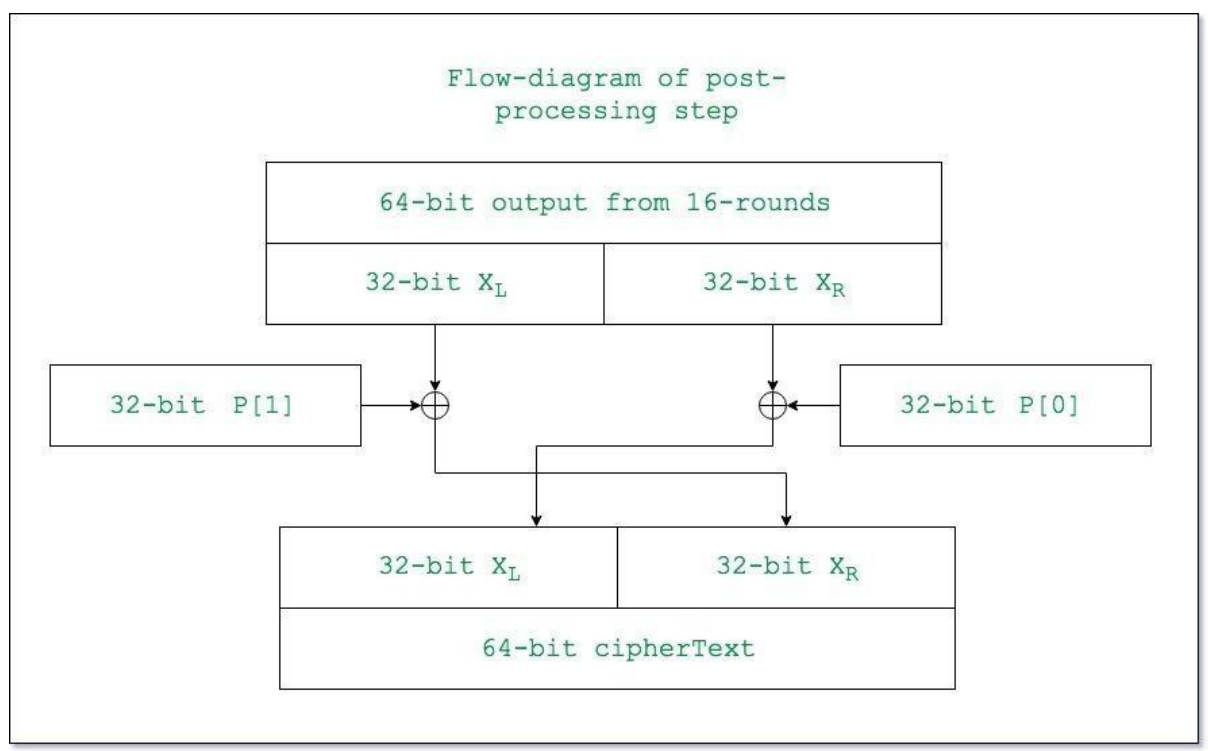
• Blowfish

It is a symmetric block cipher algorithm encrypts block data of 64-bits at a time. It will track the Feistel network. The block size used is of 64- bits and key size can vary from any length to 448. Blowfish cipher uses 18 sub arrays each of 32-bit commonly known as P-boxes and four Substitution boxes each of 32-bit, each having 256 entries.

Blowfish is a symmetric, 64-bit block cypher with a changeable length. It was designed to be a quick, free, drop-in replacement for the Data Encryption Standard (DES) and International Data Encryption Algorithm (IDEA) encryption methods, which were becoming obsolete.

Blowfish is far quicker than DES and IDEA, plus it is unpatented and freely accessible for all purposes. However, because to its tiny block size, which is deemed unsafe, it could not totally replace DES.

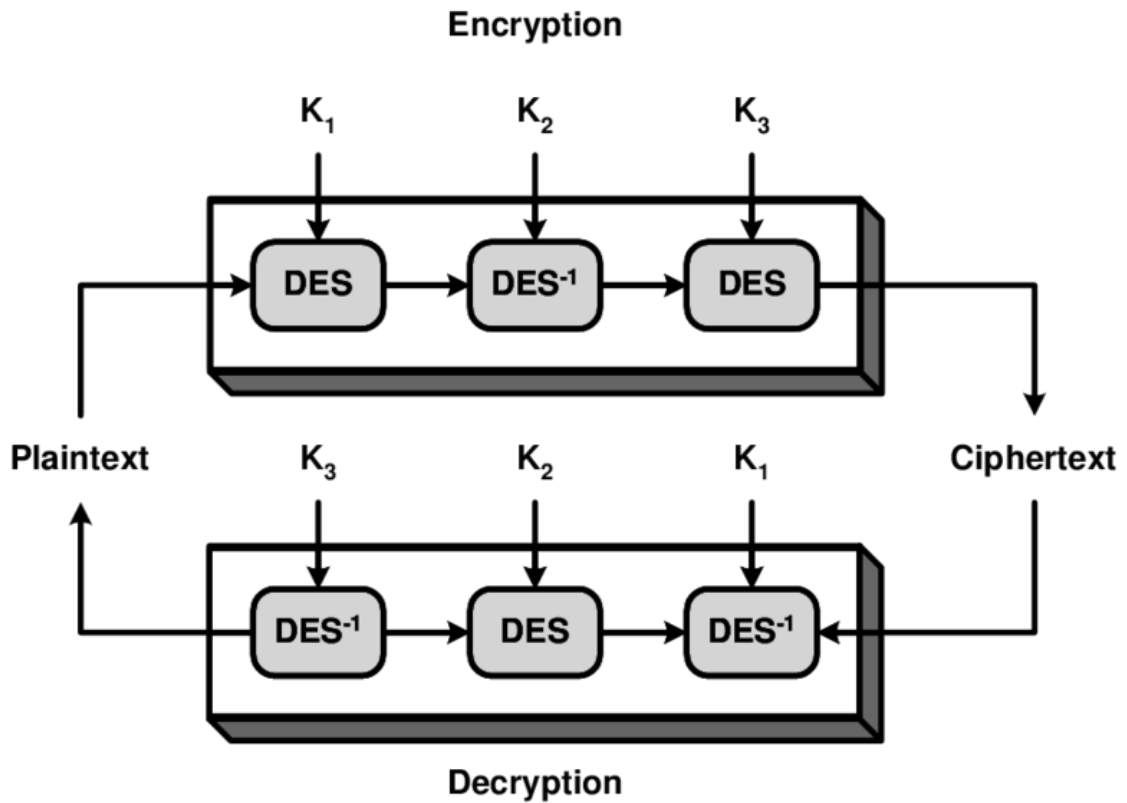
Blowfish uses a 64-bit block size and a key length that ranges from 32 to 448 bits. It is made up of 16 Feistel-like iterations, each of which works with a 64-bit block divided into two 32-bit words. Blowfish encrypts and decrypts data using the same encryption key.



• Triple DES

Another DES operating mode is triple DES. Three 64-bit keys are required for a total key length of 192 bits. Instead than inputting each of the three keys separately, you just type in the full 192-bit (24 character) key in Stealth. The Triple DES DLL then divides the user-supplied key into three subkeys, padding them as needed to make them all 64 bits long.

The encryption technique is identical to standard DES, except it is done three times, therefore the name Triple DES. The first key encrypts the data, the second key decrypts it, and the third key encrypts it again. Runs three times slower than DES but is much more secure if used properly. The procedure for decrypting something is the same as the procedure for encryption, except it is executed in reverse.



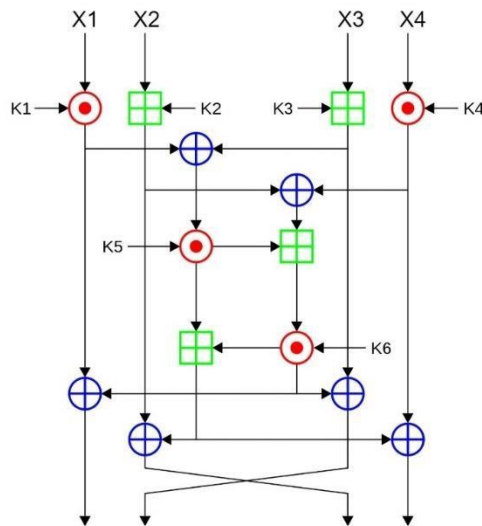
• IDEA

It is one of the ciphers which encrypt the text into an unreadable format and makes it secured to send it over to internet. The IDEA encryption algorithm provides high level security not based keeping the algorithm a secret, but rather upon ignorance of the secret key.

The IDEA algorithm (International Data Encryption Algorithm) is a type of encryption algorithm. It's a symmetric block cypher that uses a 64-bit input, a 28-bit key, and eight identical encryption rounds with six distinct sub-keys, as well as four keys for output transformation.

A common block size is 16 bytes with a 128-bit length. A block cypher usually works in round blocks, where a portion of the key is applied to the round before additional operations are done on it. We end up with our ciphertext for that block after a given number of rounds, say 10 to 16.

International Data Encryption Algorithm(IDEA)



Where,



= Modular Addition



= Modular Multiplication



= Bitwise XOR



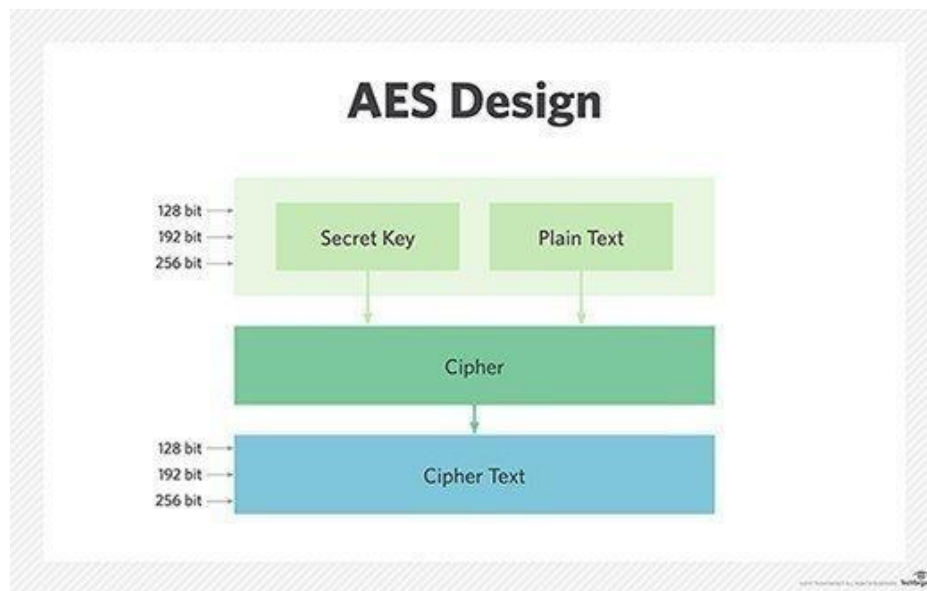
• AES

It is an iterative rather than Feistel cipher. It is based on 'substitution-permutation network'. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Rather than being a Feistel cypher, AES is an iterative cypher. It uses a 'substitution-permutation network.' It consists of a sequence of connected processes, some of which require substituting specified outputs for inputs (substitutions) and others involving shuffling bits about (permutations).

Surprisingly, AES uses bytes rather than bits for all of its calculations. As a result, AES considers a plaintext block's 128 bits as 16 bytes. For matrix processing, these 16 bytes are organised into four columns and four rows.

In contrast to DES, the number of rounds in AES is configurable and dependent on the key length. For 128-bit keys, AES employs 10 rounds, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys.



• **Fernet**

Fernet is a symmetric encryption/decryption system based on industry standards. It also authenticates the message, allowing the recipient to determine whether or not the message has been tampered with in any manner.

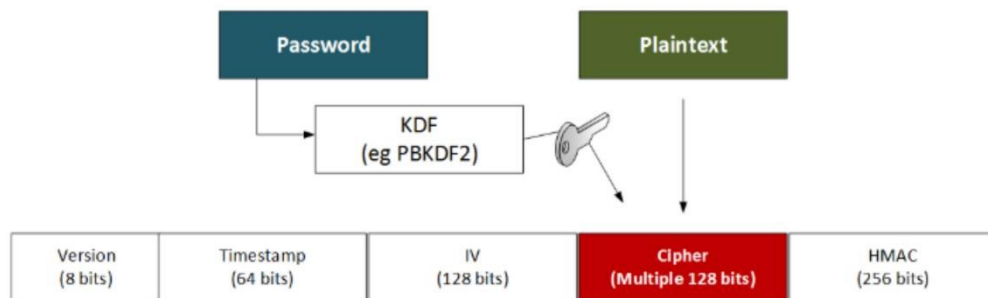
It guarantees that a message encrypted using it cannot be manipulated or read without the key. Fernet is an implementation of symmetric (also known as "secret key") authenticated cryptography. Fernet also has support for implementing key rotation via MultiFernet. The fernet module guarantees that data encrypted using it cannot be further manipulated or read without the key.

Fernet ensures that without the key, a communication encrypted with it cannot be altered or read. Fernet is a symmetric (or "secret key") authenticated cryptography system.

Fernet solves many of the challenges that an inexperienced developer would encounter while developing such a system by:

- Providing a safe method of key generation (a key is similar to a password).
- Choosing a safe encryption algorithm (AES using CBC mode and PKCS7 padding)
- To make the encryption more secure, a secure "salt" value IV) is assigned at random.
- The encrypted communication is timestamped.
- To identify any efforts to modify the message, it is signed (using HMAC and SHA256).

Fernet (symmetric encryption)



Literature Survey / Related Works

<u>Title</u>	<u>Authors</u>	<u>Methodology</u>	<u>Gaps</u>
A unique message encryption technique based on enhanced blowfish algorithm. (2019)	Dulla, G. L., Gerardo, B. D., & Medina	In this study, the concept follows the reverse, swapping and shifting of plaintext to its binary equivalent. A series of steps have been implemented to produce the encrypted values. In the encryption and decryption process,	Though the enhanced algorithm is faster than the classic Blowfish algorithm, it depends on the filesize, that is, it delivers an improved performance in terms of smaller file size. The average time difference between
		the series of steps follows the shifting, XOR and switching scheme. In the proposed study, the shift method will be used in each block directed to the function with the Sbox.	the enhanced algorithm and the classic is about 11%. The difference increases for smaller files as compared to large ones.

Enhancing the data security in Cloud by implementing hybrid (RSA & AES) encryption algorithm. (2014)	V. S. Mahalle and A. K. Shahade	A hybrid cryptography algorithm is used to ensure data security. Three separate keys are used for encryption as well as decryption. When the user wants to access or download the data, it goes through the download procedure whereby the user has to specify the filename to be downloaded and has to provide the AES and RSA key.	Though AES is highly secure, in some cases, it is complex to implement taking both performance and security into consideration
Analysis of Encryption Algorithms (RSA, SRNN and 2 Key Pair) for Information Security (2017)	S. Y. Bonde and U. S. Bhadade	RSA, short range natural number (SRNN) and two key pair algorithms are compared considering parameters encryption/decryption time. It involves three important steps: 1.] Key Generation 2.] Encryption Process 3.] Decryption Process	This paper compared the performance of three different algorithms. Though it gives a great insight as to which algorithm is better in which respect, the algorithms are tested solo. Hence, we cannot comment on the performance of hybrid algorithms using this study.
Enhancing the security of cloud data using hybrid encryption algorithms. (2019)	Sajay, K.R., Babu, S.S. & Vijayalakshmi Y	Homographic encryption and Blowfish encryption are combined to enhance cloud security. Python software tool and cryptography technique is used to enhance the cloud	Blowfish algorithm is used for developing the security and privacy issues in the cloud. It generates the key for security and a symmetric key block is used for both decryption and encryption. The

		<p>security. It is a multilayer cryptography algorithm with homographic encryption in the first layer and blowfish encryption in the second layer. The first layer of homographic encryption is applied to the input text. Then the encryption result will be obtained. After that the result of encryption is passed to the second layer which is the blowfish encryption layer. The final output of the encryption layer is obtained.</p>	<p>performance of Blowfish algorithm is proportional inversely to the size of key and if the size of key will increase then the performance will decrease and vice versa.</p>
Optimized Public Auditing and Data Dynamics for Data Storage Security in Cloud Computing (2016)	Anirudha Pratap Singh, Syam Kumar Pasupuleti	<p>They devised a strategy based on two sequential data auditing methods. To begin, the client requests data blocks from the cloud storage server during the dynamic system update process (CSS). The CSS verifies that the data in this block is correct and that the preceding change was successful. The third party then begins the "Third Party Audit (TPA) process," which is a public auditing procedure. The TPA method is based on verifying the CSS's probabilistic evidence of integrity.</p>	<p>The main security issue is identifying the TPA as an employee, which may become a malicious insider even if it is a weak possibility.</p>
Cloud Application Security using Hybrid Encryption (2020)	Alabi Orobosade, Thompson Aderonke Favour-Bethy, Alese Boniface Kayode, Arome J. Gabriel	<p>The best approach for enhancing high-level security of cloud data is to utilize hybrid encryption that uses</p>	<p>The cipher-text size to plain-text in the proposed hybrid encryption algorithm</p>

		both symmetric and asymmetric algorithms. A hybrid encryption scheme was developed to ensure data privacy. The suggested model uses the AES method with ECC key encryption, taking advantage of its properties as a fast symmetric scheme and less computationally complicated resilient cryptosystem techniques.	compared to AES and ECC is only average
Securing Data and Reducing the Time Traffic Using AES Encryption with Dual Cloud (2019)	P. Sivakumar; M. NandhaKumar; R. Jayaraj; A.Sakthi Kumaran	On this suggested effort, they applied the AES technique for data protection in the Heroku cloud (which is a cloud platform). The AES cryptography may be used to secure data on a cloud platform. Also, if one or both clouds are active, a dual cloud can be used. If a cloud is operational, data should be more efficient when it comes to uploading and downloading operations in the cloud.	Releases from using AES require well-built security from third parties where the key used in the encryption process is the same as the decryption process. This issue may make it easy to break into the original data and retrieve it.
Enabling Cloud Database Security Using Third Party Auditor (2019)	S Pandiaraj, Aishwarya, Surbhi, Alisha Minj, Priyanshu Singh	In this paper, they have proposed a personality based information trustworthiness examining instrument for secure distributed storage, which bolsters information offering to concealing the basic data. In their instrument,	The max size for files is between 100KB and 100MB

		the information put away into the cloud is capable to become public and utilized by anyone relying onto the prerequisite that the touchy data of the document was ensured.	
A New Hybrid Automated Security Framework to Cloud Storage System (2021)	Wael A. Awad, Doaa S. El-Morshedy, Noha E. El-Attar	The proposed automatic cryptographic system aims to curtail the role of the CTP by developing an Automated Encryption/Decryption System for Cloud Data Storage (AEDS), which adopts a fully automated strategy for data encryption and decryption processes. The AEDS is based on transferring all the security operations on the data to an autonomous system, beginning with uploading the data by the user on the cloud platform, passing through encrypting it to be ready for storage, and ending with decrypting it when its owner asks to retrieve it. AEDS is a hybrid cryptography framework that is implemented based on four encryption algorithms; Twofish , Advance Encryption Standard (AES) , and Data Encryption Standard (DES) as symmetric encryption algorithms , and	Time Taken for the encryption and decryption process is huge. DES algorithm can be very slow in cases where large data needs to be encrypted by the same computer

		Rivest–Shamir–Adleman (RSA) as an asymmetric encryption algorithm .	
Data Security and Privacy Protection for Cloud Storage (2020)	PAN YANG; NAIXUE XIONG; JINGLI REN	In this paper, they make a comprehensive review of the literatures on data security and privacy issues, data encryption technology, and applicable countermeasures in cloud storage system. Specifically, they first make an overview of cloud storage, including definition, classification, architecture and applications. Secondly, they give a detailed analysis on challenges and requirements of data security and privacy protection in cloud storage system. Thirdly, data encryption technologies and protection methods are summarized. Finally, discuss several open research topics of data security for cloud storage.	No novel approach was introduced in this paper rather comparing existing to analyse efficiencies of each
A new lightweight cryptographic algorithm for enhancing data security in cloud computing (2021)	Fursan Thabit; Sharaf Alhomdy; Abdulrazzaq H.A.Al-Ahdal; Prof Dr Sudhi Jagtap	This research proposes a New Lightweight Cryptographic Algorithm for Data Security Enhancement that may be utilized to safeguard cloud computing applications. The algorithm is a 16-byte	Using a single algorithm (non-hybrid methodologies) is insufficient to provide high levels of security because, in symmetric algorithms, only one encryption key is used to encrypt and decrypt data.

		<p>(128-bit) block cipher, and the data must be encrypted using a 16byte (128-bit) key. It is based on feistel and replacement permutation architectural approaches to increase encryption difficulty. Shannon's notion of dispersion and confusion is achieved by the use of logical processes such as (XOR, XNOR, shifting, swapping). The length of the secret key and the number of spins are also adjustable. When compared to existing cryptographic systems, the experimental findings of the proposed method showed a high degree of security and an evident improvement in cipher execution time and security forces.</p>	
Enhancing Cloud Data Security using Multilevel Encryption Techniques Enhancing Cloud Data Security using Multilevel Encryption Techniques (2021)	Najd Almoysheer , Mamoonah Humayun , A. A. bd El-Aziz , NZ Jhanjhi	<p>The proposed method aim to protect data exchanged between server and client n SaaS. The proposed technique model is a hybrid method that combines both the symmetric and asymmetric cryptography techniques. They proposed uses use the Blowfish encryption to encrypt could data and Elliptic Curve Cryptography (ECC) to generate and manage</p>	<p>This is an improved system which has high security with great performance but increased encryption and decryption time.</p>

		encryption keys. It is a quite important task to protect data stored on cloud.	
A unique message encryption technique based on enhanced blowfish algorithm (2019)	Godfrey L Dulla, Bobby D Gerardo, and Ruji P Medina	This study aims to secure both plaintext and file message content through encryption technique which is based on enhanced Blowfish algorithm. An enhanced Blowfish algorithm is developed to improve its performance by reducing the number of rounds and by increasing the block length with fixed length during encryption and decryption with added transformation method on selected rounds	Using a single algorithm (non-hybrid methodologies) is insufficient to provide high levels of security because, in symmetric algorithms, only one encryption key is used to encrypt and decrypt data.
Secure File Storage Using Hybrid Cryptography (2020)	Aditya SadanandGhadi	This discussed paper is a broad survey of the different approach which is used for securely storing files, and sharing it over the network. This proposed scheme will also ensure the whole model to have confidentiality, integrity, and availability mechanisms to be implemented in it.	One of the disadvantages of the current system is that it uses steganography to share secret keys between users which.

Secure File Storage Using Hybrid Cryptography (2020)	Ronak Karani, Tejas Choudhari, Anindita Bhajan, Madhu Nashipudimath	The following paper is a broad survey of different approaches that can be used to securely store a file over a network and share it. Accordingly, a scheme is proposed for the same. The	One of the disadvantages of the current system is that it uses steganography to share secret keys between users which.
--	---	--	--

		proposed scheme will provide a way for storing files securely using cryptography ensuring confidentiality, integrity and authentication security mechanisms	
Secure File Storage on Cloud Using Hybrid Cryptography Algorithm (2020)	Uttam Kumar, Mr. Jay Prakash	The main aim of this system is to securely store and retrieve data on the cloud that is only controlled by the owner of the data. Cloud storage issues of data security are solved using cryptography and steganography techniques. Data security is achieved using RC6, 3DES and AES algorithm. Key information is safely stored using LSB technique (Steganography).	The proposed system after multithreading still takes a lot of time and resources to encrypt and decrypt.

Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing (2013)	Prashant Rewagad; Yogita Pawar	In this paper, they have proposed to make use of digital signature and Diffie Hellman key exchange blended with (AES) Advanced Encryption Standard encryption algorithm to protect confidentiality of data stored in cloud. Even if the key in transmission is hacked, the facility of Diffie Hellman key exchange render it useless, since key in transit is of no use without user's private key, which is confined only to the legitimate user. This proposed	Time consuming procedure as three different steps using different techniques are performed
---	-----------------------------------	--	--

		architecture of three way mechanism makes it tough for hackers to crack the security system, thereby protecting data stored in cloud.	
--	--	---	--

Secure File Storage & Sharing on Cloud Using Cryptography (2021)	Madhumala RB ; Sujan Chhetri ; Akshatha KC ; Hitesh Jain	In this paper, the plan proposed is to overcome the issues regarding the data that are being stored by the users on the cloud should be encrypted rather than storing them in a plain form such that the data will be protected from the attackers who are trying to read, delete or manipulate the data. The application is focused on securely authenticating the user, before storing and sharing files, To create an application that lets a user encrypt and decrypt any type of file without any changes in the size during encryption & decryption, store every user data in the encrypted form on the cloud, to provide a communication medium between users via the chat application, to give direct access to the file for CRUD operation only to the owner.	In the proposed system, the files encrypted during the testing phase are very small therefore not accurate as size increases.
Efficient and finegrained sharing of encrypted files(2010)	Songling Fu; Xiang-Ke Liao; Lianyue He; Chenlin Huang	In this work, they present an efficient fine-grained sharing	There is Data leakage problem, there is an

		<p>approach of encrypted files. A concept named safe capsule (SC) is proposed as the organization unit for files. By using safe capsule, user can provide one of the finegrained access permissions of their own data to others, such as read-only</p>	<p>increase in the decrypted file size and encryption methods are limited to file types</p>
<p>Secure File storage in Cloud Computing using Hybrid Cryptography Algorithm(2018)</p>	<p>Punam V.Maitri ; Arun Verma</p>	<p>In this proposed system AES, blowfish, RC6 and BRA algorithms are used to provide block wise security to data. All algorithm key size is 128 bit.LSB steganography technique is introduced for key information security. Key information contains which part of file is encrypted using by which algorithm and key. File is split into eight parts. Each part of file is encrypted using different algorithm. All parts of file are encrypted simultaneously with the help of multithreading technique. Data encryption Keys are inserted into cover image using LSB technique. Stego image is send to valid receiver using email .For file decryption purpose reverse process of Encryption is applied.</p>	<p>RC6 is not universally practical and maximum delay is needed for data encode and decode</p>

Secure File storage on Cloud Computing using Hybrid Cryptography Algorithm (2020)	Aishwarya S. Dashmukhe ; Nilesh Alone	In this paper they introduce a new cloud computing security by using Symmetric key cryptography algorithm and stenography. In proposal system AES, Blowfish, RC6, BRA algorithms are included for security purpose. All algorithms have 128bit key size and file divide into Eight parts.	RC6 is not universally practical and maximum delay is needed for data encode and decode
Cloud Security Solutions: Comparison among Various Cryptographic Algorithms (2018)	Jagriti Dhamija	This study examines three different algorithms: AES, DES, and ECC. They present numerous details on how the algorithm works and how effective it is when applied to real-world challenges.	This paper does not introduce a new system or a hybrid rather analyses the current algorithms and gives a report on how each fare in realtime.
Review of Secure File Storage on Cloud using Hybrid Cryptography (2020)	Shruti Kanatt; Amey Jadhav; Prachi Talwar	This paper presents a review of a system which stores data on the cloud after encrypting it. Hence even if a security breach were to take place, the attacker would get access to encrypted data, which would still ensure data confidentiality. In this system, the user uploads a file to the portal, it gets encrypted and then uploaded onto the cloud. The user can then download their files from the cloud through the portal, which results in the decrypted (or original) file getting	The Blowfish algorithm is utilized in the cloud to develop security and privacy challenges. A symmetric key block is utilized for both decryption and encryption, and it creates the key for security. The performance of the Blowfish method is inversely related to the size of the key; as the key size grows, so does the performance, and vice versa.

		downloaded to their local computer.	
Survey Paper on Cloud Storage Security(2013)	Sunita Sharma,Amit Chugh	Using EFS, NTFS with cache for securing data files by using automatic cryptographic systems inbuilt in EFS.	As cryptographic systems are inbuilt in EFS, modifications for providing better security measures is difficult to implement.
Secure File Storage and File Sharing (2017)	Rawal, B. S. ; Vivek, S. S.	Separate servers are used for input, storage and output functions. Providing better security by keeping separate modules.	As three different servers are used there can be connectivity issues as well as synchronization problems
A novel data classification-based scheme for cloud data security using various cryptographic algorithms (2021)	Narender Kumar; Mohd Naved Ul Haq	According to the approach to encryption and decryption, the proposed work uses four algorithms and their pairs such as AES-256 +DES, AES-256+3DES, AES-256+Blowfish cascade cipher. So, all of these cannot be reasonably compared to each other. Therefore, they are all analysed separately. The experiments were performed on files of various sizes such that they reasonably covered files of various sizes. To better analyze the results, files are treated in two categories based on their sizes	For small files the execution time is linear but as size increases , execution time increases linearly.

Overall Architecture

In this project we will be splitting the project into 3 parts:

☒ **Web app development.**

This section will be the interface in which the user will be interacting with. Here the user will be able to encrypt and decrypt their files. Flask provides a great platform to create web pages with integration of python files as well. This web app will be available to the masses and depending on the key the user has received can the user retrieve back the encrypted file back to its normal state.

☒ **Encryption Decryption Phase**

In the proposed algorithm, we will have two main phases - Encryption Phase and Decryption Phase. In the encryption phase, we will split the file into pieces and encrypt it with encryption algorithms like AES, Blowfish, Triple DES, IDEA, Fernet. Each split will be encrypted in a separate thread making the process faster and more optimized. We will obtain a secured key that the user can download. In the Decryption phase, we will provide the secured key, which will be used to decrypt the split files stored in the server. After uploading the key, it will decrypt the different splits using the respective decryption algorithms and finally the user will be given the option to download the decrypted text that would match the original file uploaded.

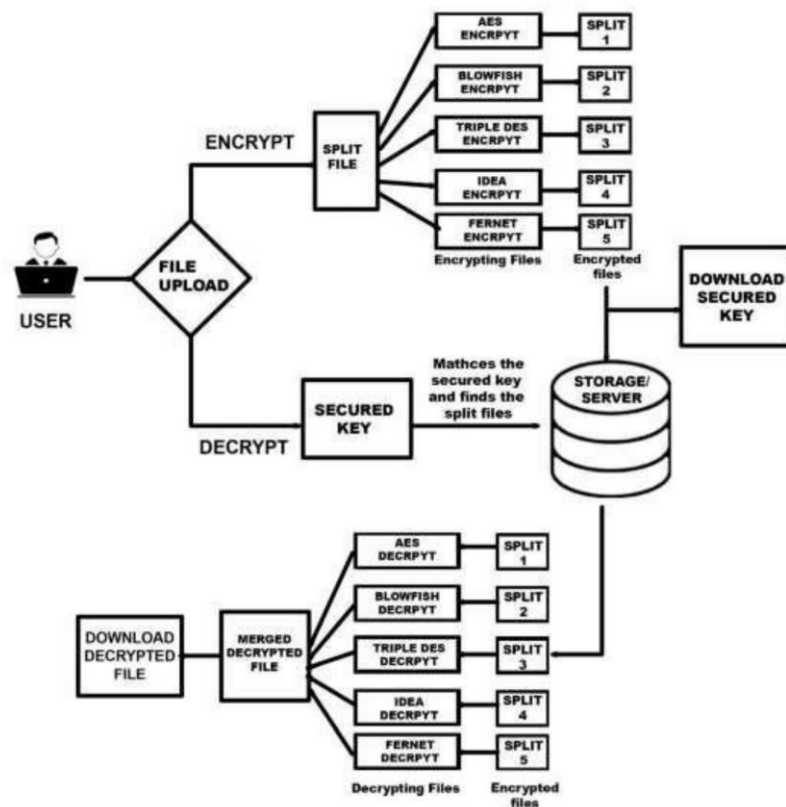
☒ **Threading**

When a process has many jobs to complete independently of one another, threads are particularly useful in contemporary programming. This is especially true when one of the jobs may block and the other duties must continue without being obstructed. In this case threading plays a huge role to decrease the amount of time it takes to encrypt and decrypt the file. Proper threading will make the speed of our system equal or even better than the other algorithms found in the industry.

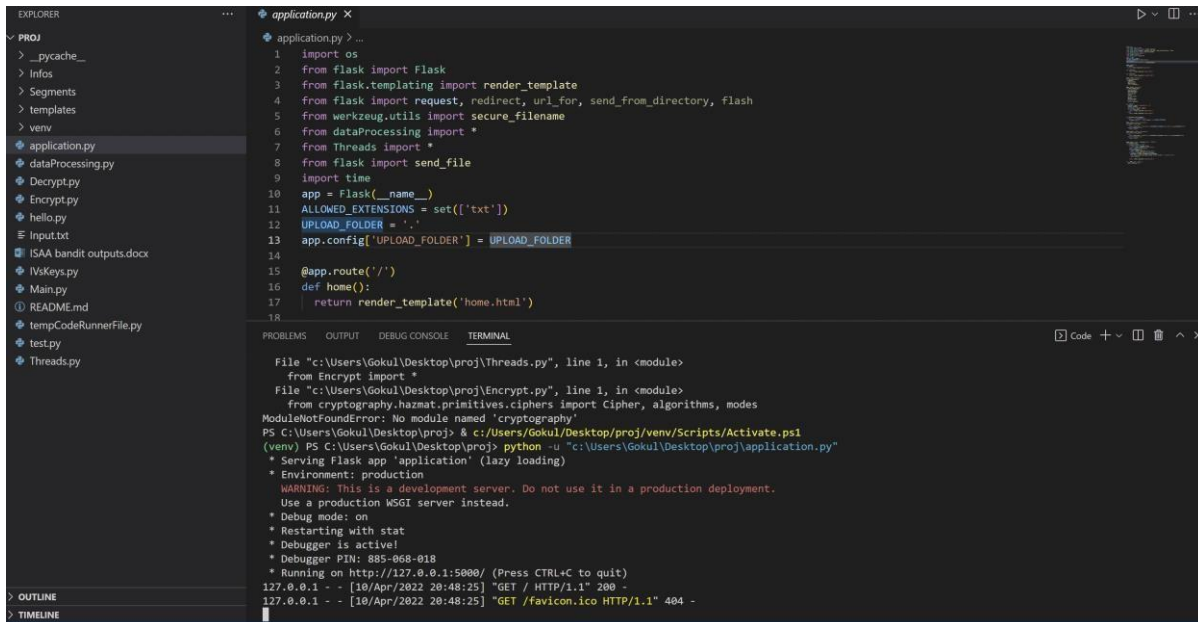
Proposed Methodology

In the proposed algorithm, we will have two phases - Encryption Phase and Decryption Phase. In the encryption phase, we will split the file into pieces and encrypt it with encryption algorithms like AES, Blowfish, Triple DES, IDEA, Fernet. Each split will be encrypted in a separate thread making the process faster and more optimized. We will obtain a secured key that the user can download.

In the Decryption phase, we will provide the secured key, which will be used to decrypt the split files stored in the server. After uploading the key, it will decrypt the different splits using the respective decryption algorithms and finally the user will be given the option to download the decrypted text that would match the original file uploaded.



Results

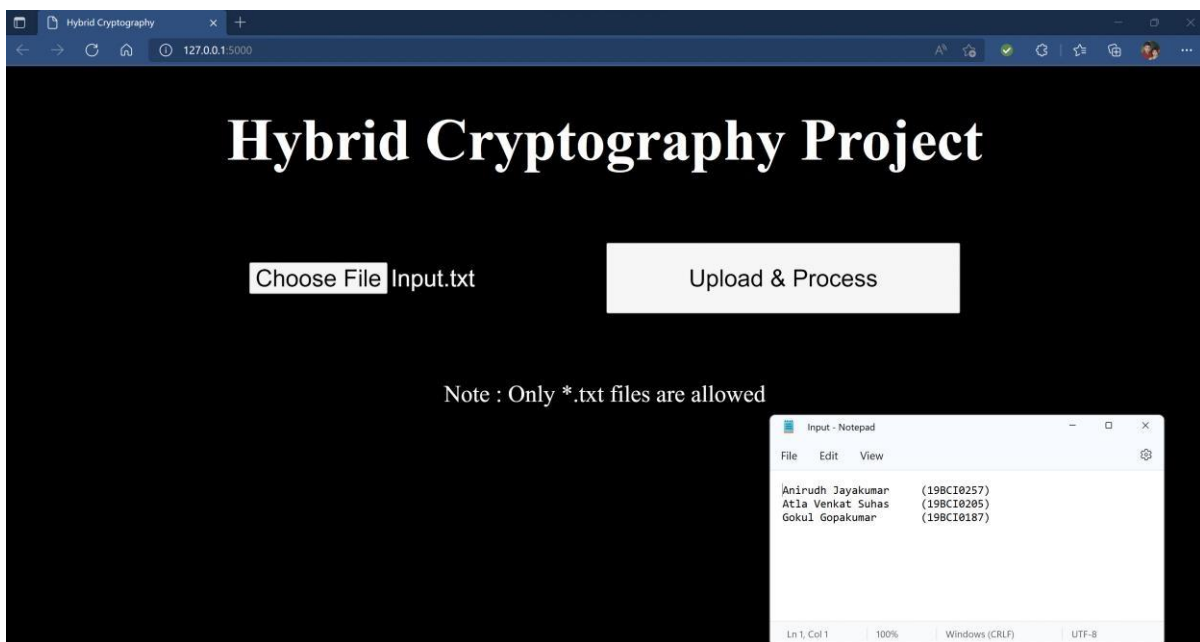


The screenshot shows a VS Code editor with a file explorer on the left containing files like `_pycache_`, `Infos`, `Segments`, `templates`, `venv`, `application.py`, `dataProcessing.py`, `Decrypt.py`, `Encrypt.py`, `hello.py`, `Input.txt`, `ISAA bandit outputs.docx`, `IVsKeys.py`, `Main.py`, `README.md`, `tempCodeRunnerFile.py`, `test.py`, and `Threads.py`. The main editor displays `application.py` with the following code:

```
1 import os
2 from flask import Flask
3 from flask.templating import render_template
4 from flask import request, redirect, url_for, send_from_directory, flash
5 from werkzeug.utils import secure_filename
6 from dataProcessing import *
7 from threads import *
8 from flask import send_file
9 import time
10 app = Flask(__name__)
11 ALLOWED_EXTENSIONS = set(['txt'])
12 UPLOAD_FOLDER = '.'
13 app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
14
15 @app.route('/')
16 def home():
17     return render_template('home.html')
```

The terminal at the bottom shows the following output:

```
File "c:\Users\Gokul\Desktop\proj\Threads.py", line 1, in <module>
  from Encrypt import *
File "c:\Users\Gokul\Desktop\proj\Encrypt.py", line 1, in <module>
  from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
ModuleNotFoundError: No module named 'cryptography'
PS C:\Users\Gokul\Desktop\proj> & c:\Users\Gokul\Desktop\proj\venv\Scripts\Activate.ps1
(venv) PS C:\Users\Gokul\Desktop\proj> python -u "c:\Users\Gokul\Desktop\proj\application.py"
* Serving Flask app 'application' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 885-068-018
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [10/Apr/2022 20:48:25] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [10/Apr/2022 20:48:25] "GET /favicon.ico HTTP/1.1" 404 -
```





Hybrid Crypto!!

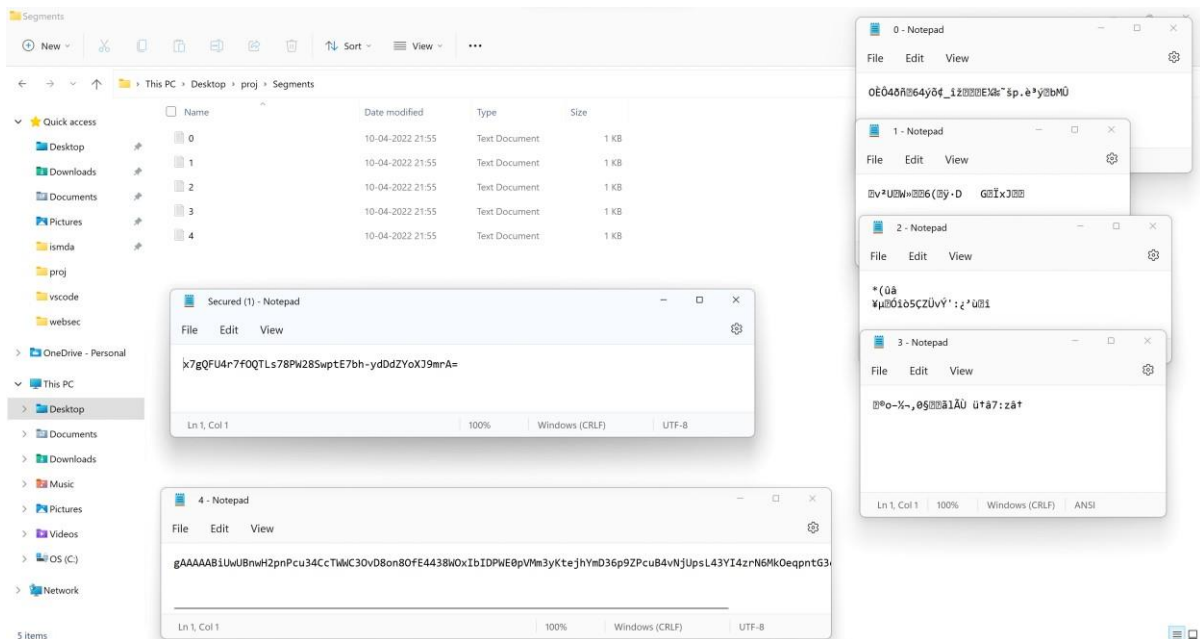
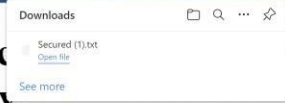
Encrypt!

Decrypt!



Thank You For Using Our Service
Your File Is Saved Successfully.

Download Key!



Segmentation and the Encrypted files





Hybrid Crypto!!!

Invalid File Format !

Only *.txt files allowed.

Choose File No file chosen

Upload & Process



Hybrid Crypto!!!

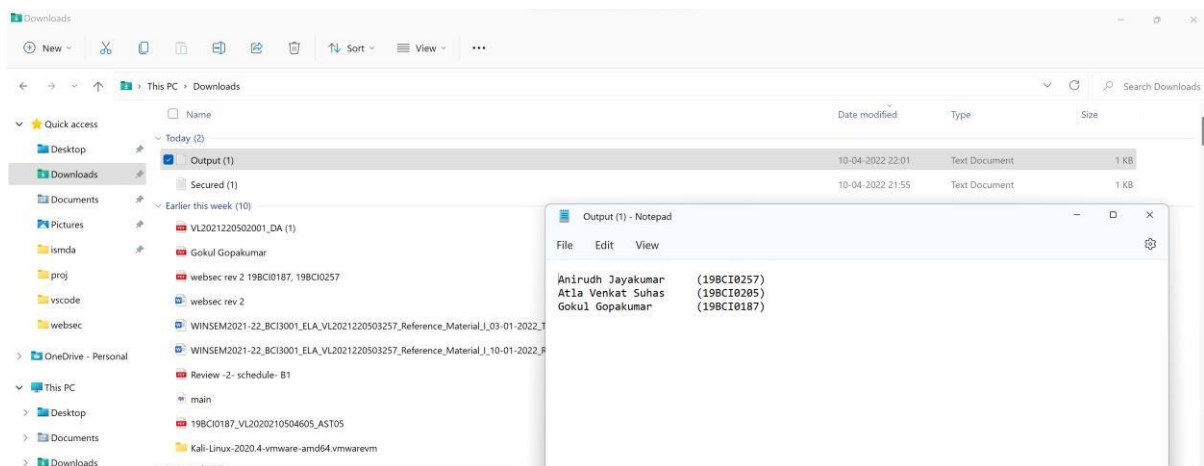
No file selected !

Only *.txt files allowed.

Choose File No file chosen

Upload & Process

Decryption



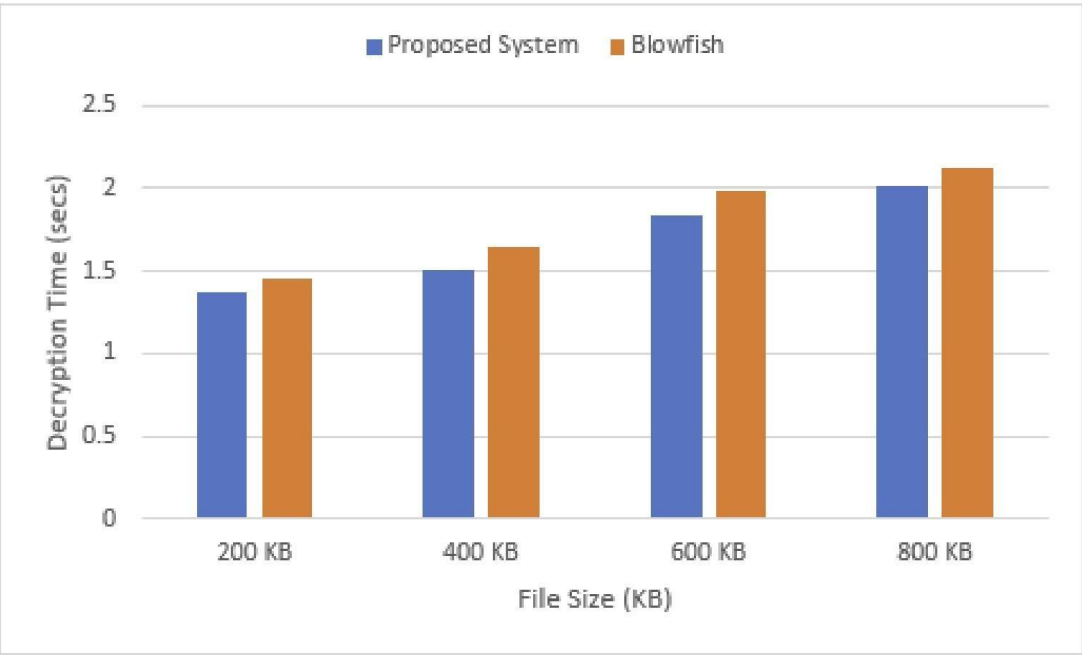
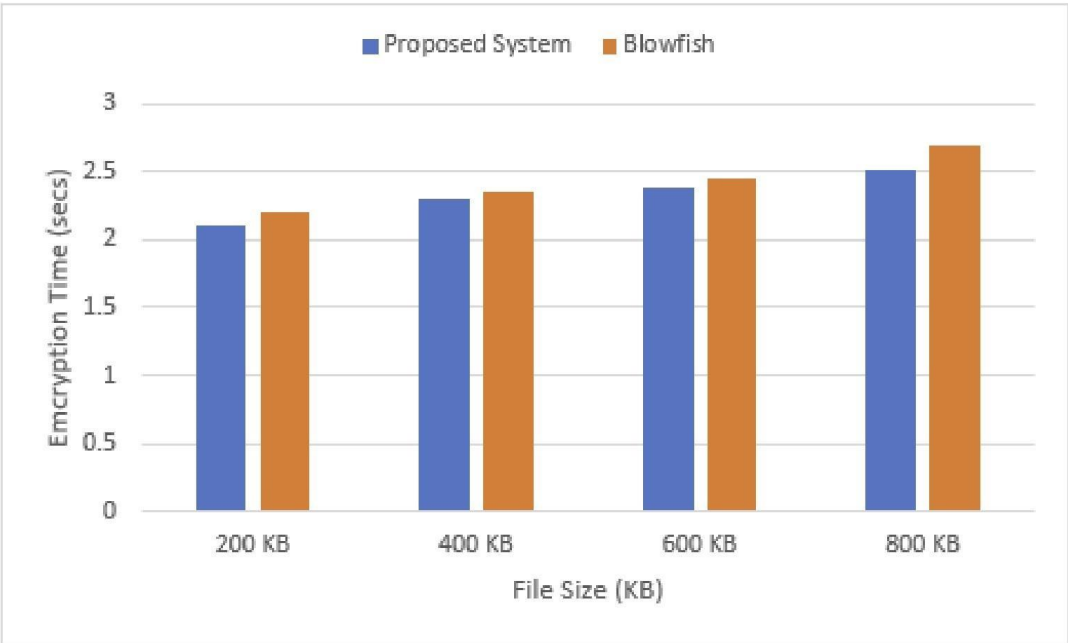
Analysis

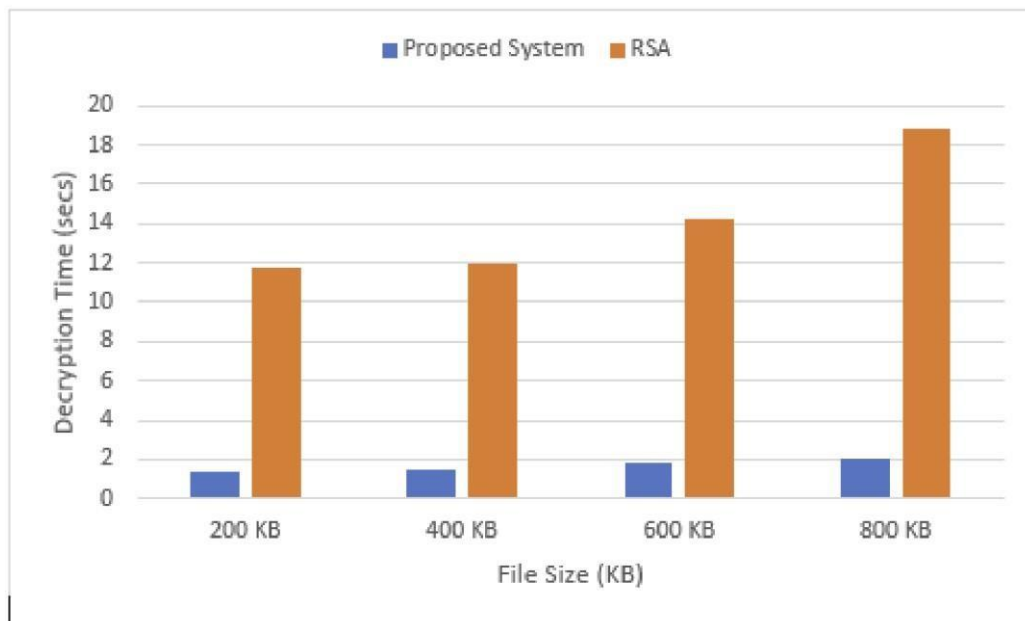
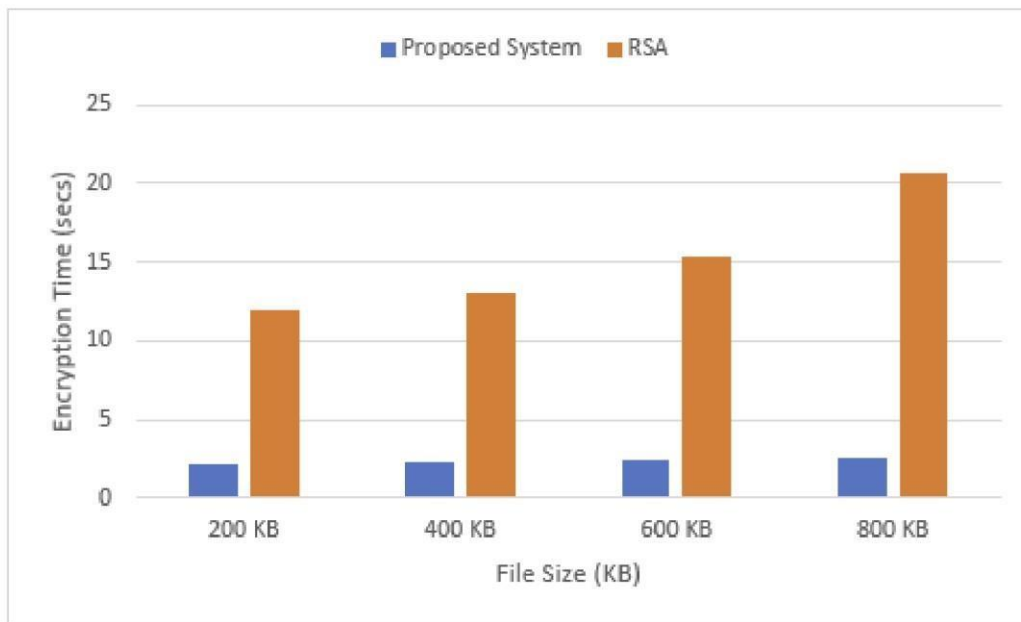
Performance

The following graphs have been used to evaluate the performance of the proposed hybrid system compared with conventional encryption algorithms like RSA and Blowfish algorithm. The parameters used for this are:

- Encryption Time
- Decryption Time

All the evaluation has been done on .txt files with file size of 200, 400, 600 and 800 KB.





The graphs show that there is negligible difference in the encryption and decryption times of proposed System and Blowfish System, but since storing the file as multiple splits encrypted with multiple algorithms make it unfeasible to attack, there is a significant difference in RSA, due to big values of prime numbers and the random number generators. The use of multithreading where each thread runs a different encryption algorithm make it faster and efficient.

Vulnerabilities in our Python environments are really irritating. They slow us down, are difficult to identify, and can delay the development process. What makes them more frustrating is that most of these vulnerabilities are known. Countless developers would have faced them before, and they report these issues online.

[illegible]

```

                                     /$$$$$          /$$
                                   /$$_ $$          | $$
    /$$$$$$ /$$$$$ | $$ \_ /$$$$$ /$$$$$ /$$ /$$
  /$$_ /  |  _ $ $ $$$$ /$$_ $$ |  _ $ /  |  $ $
|  $$$$ /$$$$$ $ $ /  |  $$$$ |  $ $ |  $ $
  \_ $ $ /$ _ $ $ $ $ |  $ _ /  |  $ /$ $ $ $
 /$$$$$/ |  $$$$ $ $ |  $$$$ |  $$$/  $$$$$$
|  _ /  |  _ /  |  _ /  |  _ /  |  _ $ $
                                     /$ $ |  $ $
                                     |  $$$$ /
                                     \_ /

by pyup.io

```

```

REPORT
checked 63 packages, using free DB (updated once a month)

```

package	installed	affected	ID
werkzeug	2.0.1	<2.0.2	42050

```

Werkzeug version 2.0.2 improves the security of the debugger cookies.
"SameSite" attribute is set to "Strict" instead of "None", and the secure
flag is added when on HTTPS.

```

Conclusion

Through this project we were successful in developing a secure file storage system. The proposed model is liable to meet the required security needs of data center of cloud. This can help maintain the integrity of cloud storage, protect unaware or vulnerable people storing files online.

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. For data security and privacy protection issues, the fundamental challenge of separation of sensitive data and access control is fulfilled.

The various benefits are as summarized:

The public key cryptography used helps to facilitate authorization of user for each file.

The need of more light and secure encryption system for file information preserving system on cloud is satisfied.

The file splitting and merging makes the model unfeasible to get attacked.

Future Work

There can be further enhancements to the project done, such as :

- ✚ Working with more features, being available on more platforms.
- ✚ Log all the details properly for further analysis of the data transfer and errors.
- ✚ Regular analysis of the website so that no flow is remained and Limit access to administrator interfaces and the implementation of the policy should also be reviewed via regular audits.

This can help reduce the chances of user data leak and provide more security to the users

References

- Mahalle, V. S., & Shahade, A. K. (2014). Enhancing the data security in Cloud by implementing hybrid (Rsa & Aes) encryption algorithm. 2014 International Conference on Power, Automation and Communication (INPAC), Power, Automation and Communication (INPAC), 2014 International Conference On, 146–149.
<https://doi.org/10.1109/INPAC.2014.6981152>
- Bonde, S. Y. (1), & Bhadade, U. S. (2). (n.d.). Analysis of Encryption Algorithms (RSA, SRNN and 2 Key Pair) for Information Security. 2017 International Conference on

Computing, Communication, Control and Automation, ICCUBEA 2017.

<https://doi.org/10.1109/ICCUBEA.2017.8463720>

- Sivakumar.P, NandhaKumar, M., Jayaraj, R., & Kumaran, A. S. (2019). Securing Data and Reducing the Time Traffic Using AES Encryption with Dual Cloud. 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), System, Computation, Automation and Networking (ICSCAN), 2019 IEEE International Conference On, 1–5. <https://doi.org/10.1109/ICSCAN.2019.8878749>
- Rewagad, Prashant & Pawar, Yogita. (2013). Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing. 437-439. 10.1109/CSNT.2013.97.
- R B, Madhumala & Chhetri, Sujana & KC, Akshatha & Jain, Hitesh. (2021). Secure File Storage & Sharing on Cloud Using Cryptography. International Journal of Computer Science and Mobile Computing. 10. 49-59. 10.47760/ijcsmc.2021.v10i05.005.
- S. Fu, X. Liao, L. He, C. Huang, X. Tang and S. Zheng, "Efficient and fine-grained sharing of encrypted files," 2010 IEEE 18th International Workshop on Quality of Service (IWQoS), 2010, pp. 1-2, doi: 10.1109/IWQoS.2010.5542714.
- Rewagad, P., & Pawar, Y. (2013). Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing. 2013 International Conference on Communication Systems and Network Technologies, Communication Systems and Network Technologies (CSNT), 2013 International Conference On, 437–439. <https://doi.org/10.1109/CSNT.2013.97>
- Secure File storage on Cloud Computing using Hybrid Cryptography Algorithm (2020) Aishwarya S. Dashmukhe ; Nilesh Alone
- "Cloud Security Solutions: Comparison among Various Cryptographic Algorithms", International Journal of Novel Research and Development (www.ijnrd.org), ISSN:2456-4184, Vol.3, Issue 4, page no.138-140, April-2018, Available :<http://www.ijnrd.org/papers/IJNRD1804025.pdf>
- Kanatt, S., Talwar, P., & Jadhav, A. (2020). Review of Secure File Storage on Cloud using Hybrid Cryptography. International Journal of Engineering Research and, 9
- Sharma, S., & Chugh, A. (2013). SURVEY PAPER ON CLOUD STORAGESECURITY. International Journal of Innovative Research in Computer and Communication Engineering, 1, 208-213.
- Rawal, Bharat. (2017). Secure Cloud Storage and File Sharing

- Haq, Mohd & Kumar, Narender. (2021). A novel data classification-based scheme for cloud data security using various cryptographic algorithms. International Review of Applied Sciences and Engineering. 10.1556/1848.2021.00317.
- Dulla, G. L., Gerardo, B. D., & Medina A unique message encryption technique based on enhanced blowfish algorithm (2019).
- Sajay, K.R., Babu, S.S. & Vijayalakshmi Y. Enhancing the security of cloud data using hybrid encryption algorithms. (2019)
- Anirudha Pratap Singh, Syam Kumar Pasupuleti. Optimized Public Auditing and Data Dynamics for Data Storage Security in Cloud Computing (2016).
- Alabi Orobosade, Thompson.A, Favour-Bethy, A.B.Kayode, A.J. Gabriel. Cloud Application Security using Hybrid Encryption (2020).
- S Pandiaraj, Aishwarya, Surbhi, Alisha.M, Priyanshu.S. Enabling Cloud Database Security Using Third Party Auditor(2019).
- Wael A. Awad, Doaa S.El-Morshedy, Noha E.El-Attar. A New Hybrid Automated Security Framework to Cloud Storage System (2021).
- PAN YANG; NAIXUEXIONG; JINGLI REN. Data Security and Privacy Protection for Cloud Storage (2020).
- Fursan Thabit; Sharaf Alhomdy; Abdulrazzaq H.A.Al-Ahdal; Prof Dr Sudhi Jagtap. A new lightweight cryptographic algorithm for enhancing data security in cloud computing (2021).
- Najd Almoysheer ,Mamoona Humayun ,A. A. bd El-Aziz , NZ Jhanjhi. Enhancing Cloud Data Security using Multilevel Encryption Techniques Enhancing Cloud Data Security using Multilevel Encryption Techniques (2021).
- Godfrey L Dulla, Bobby D Gerardo, and Ruji P Medina. A unique message encryption technique based on enhanced blowfish algorithm (2019).
- Aditya SadanandGhadi. Secure File Storage Using Hybrid Cryptography (2020).
- Ronak Karani, Tejas Choudhari, Anindita Bhajan, Madhu Nashipudimath. Secure File Storage Using Hybrid Cryptography (2020).
- Uttam Kumar, Mr. Jay Prakash. Secure File Storage on Cloud Using Hybrid Cryptography Algorithm (2020).

Appendix

- 19BCI0187 : Gokul Gopakumar – Threading
- 19BCI0257 : Anirudh Jayakumar - Web Development
- 19BCI0205 : Atla Venkat Suhas – Encryption & Decryption