

SECURE CLOUD STORAGE USING HYBRID CRYPTOGRAPHY

Submitted in partial fulfillment of the requirements of

University of Mumbai

For the Degree of

Bachelor of Engineering in CSE (AI & ML)

Submitted by

Ms. SHRUTI KAMBALI [ROLL NO:- 29]

Mr. GHANSHYAM GADEKAR [ROLL NO:- 16]

Ms. POORVA PADAVE [ROLL NO:- 43]

Ms. POORVA PATIL [ROLL NO:- 49]

Under the guidance of

PROF. SWATI VYAS



DEPARTMENT OF CSE (AI & ML)

SMT. INDIRA GANDHI COLLEGE OF ENGINEERING

Ghansoli, Navi Mumbai - 400701

Academic year: 2022-2023

Project Report Approval for T.E.

This project report entitled **“SECURE CLOUD STORAGE USING HYBRID CRYPTOGRAPHY”**

By

Ms. SHRUTI KAMBALI [ROLL NO:- 29]

Mr. GHANSHYAM GADEKAR [ROLL NO:- 16]

Ms. POORVA PADAVE [ROLL NO:- 43]

Ms. POORVA PATIL [ROLL NO:- 49]

Are approved for the degree of Bachelor of Engineering in Computer Engineering, Semester VI,
University of Mumbai.

Examiner 1

Examiner 2

Prof. Swati Vyas
(Project Guide)

Prof. Sonali Deshpande

Head of Department

Dr. Sunil Chavan

Principal

Date:

Place: Ghansoli, Navi Mumbai.

Declaration

We declare that this written submission represents our own ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any act/data/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Ms. SHRUTI KAMBALI [ROLL NO:-29]

Ms. POORVA PADAVE [ROLL NO:- 43]

Ms. POORVA PATIL [ROLL NO:- 49]

Mr. GHANSHYAM GADEKAR [ROLLNO:- 16]

Date:

Place: Ghansoli, Navi Mumbai.

Abstract

This proposed hybrid algorithm can be made much more powerful and secure by increasing the number of iterations in the encryption algorithm to suit the level of security required. An inverse policy of reducing the number of iterations for lower security can also be employed.

Blowfish used for the encryption of file slices takes minimum time and has maximum throughput for encryption and decryption from other symmetric algorithms.

The idea of splitting and merging adds on to meet the principle of data security. This system can be implemented into banking and corporate sectors to securely transfer confidential data in the world of data being the key asset, safeguarding our asset is primary responsibility.

Communications, databases, infrastructure, transactions, knowledge; an organization's data is arguably its most valuable asset. It is in a business' best interests to keep its information safe, regardless of legal or regulatory requirements. For data security and privacy protection issues, the fundamental challenge of separation of sensitive data and access control is fulfilled. Cryptography technique translates original data into unreadable form.

Cryptography technique is divided into symmetric key cryptography and public key cryptography. This technique uses keys for translate data into unreadable form. So only authorized person can access data from cloud server. Cipher text data is visible for all people.

It works on 9 stages:-

- 1) User Selects file
- 2) Encrypt using AES, DES, IDEA or Blowfish
- 3) Generate key
- 4) Upload same or another file
- 5) User Search File
- 6) View File & Send request
- 7) Enter Key
- 8) Decrypt using Key
- 9) Download file

Objectives of the Project

(Every purpose / feature of project should be enumerated point wise.)

1. The stored image file is completely secured, as the file is being encrypted not by just using one but five Encryption algorithm which are AES, DES, Blowfish, IDEA and RC6.
2. The key is also safe as it embeds the key in image using LSB.
3. The system is very secure and robust in nature.
4. Data is kept secured on cloud server which avoids unauthorized access.
5. A system which stores data after encryption.
 - This prevents data leak if a breach occurred.
 - Any form of data can be stored.
 - It ensures data confidentiality to users.
 - The primary goal of the system is to provide and simulate a solution to face the challenges and solve security issues that exists.

List of Abbreviations

- | B.E.: Bachelor of Engineering
- | DFD: Data Flow Diagram
- | AES:- Advanced Encryption Standard
- | DES:-Data Encryption Standard
- | IDEA:- International Data Encryption Algorithm
- | RC6:- Rivest cipher 6
- | ECC:- (Elliptic Curve Cryptography)
- | MVT Diagrams
- | VS Code: Visual Studio Code

List of Figures

- | Fig.1 VS Code
- | Fig. 2 Architecture Of System
- | Fig. 3 Block Diagram Showing Working Of System
- | Fig 4. Data Flow Diagram (Level 0)
- | Fig 5.Data Flow Diagram (Level 1)
- | Fig.6. Flow Chart
- | Fig8. Sequence Diagram
- | Fig 9. UML Diagram
- | Fig.10.ER Diagram
- | Fig 11 All Models
- | Fig 12. Starting Development Server SS.
- | Fig.13. Original File to Encrypt & Decrypt Images.
- | Fig.14. File Uploading & Selecting Page
- | Fig. 15. File Encryption Page
- | Fig 16. No File Chosen Page
- | Fig. 17 Encrypted File page

List of Tables

- | Table 3.1 Timeline Chart
- | Table 3.2 Gantt Chart

INDEX

Chapter No.	Content		Page No.
I	Abstract		4
II	List of Abbreviations		6
III	List of Figures & Tables		6
1	Introduction		10
	1.1	Problem Statement	11
	1.2	Objectives	12
	1.3	Scope	13
	1.4	Report Organization	14
2	Literature Survey		15
	2.1	Research Paper Analysis, literature survey	16
	2.2	Methodology	17
	2.2.1	Visual Studio Code (VS Code)	17-19
3	Planning and Formulation		20
	3.1	Project Development Model	21
	3.1.1	Project Model: (any process model) e.g Block Diagram of System	21
	3.1.2	Workflow of System	22
	3.1.3	Advantages of Hybrid Cryptography in Model	22
	3.1.4	Disadvantages of Hybrid Cryptography in Model	22
	3.2	Timeline Chart	23
	3.3	Gantt Chart	24
	3.3	Feasibility Analysis	25
	3.4.1	Technical Feasibility	25
	3.4.2	Economic Feasibility	25
	3.4.3	Operational Feasibility	25

4	Requirement Analysis		26
	4.1	Hardware Requirements	27
	4.2	Software Requirements	27
	4.3	Functional Requirements	27
	4.4	Non-functional Requirements	28
	4.5	Data Flow Diagrams	29
		4.5.1 DFD Level 0	29-30
		4.5.2 DFD Level 1	31
5	System Design		32
	5.1	Flow Chart	33
		5.2 Process Flow Diagram	34
		5.3 Sequence Diagram	35
		5.4 System Architecture	36
		5.5 UML Diagram	37
		5.6 System process Diagram	38
		5.7 ER Diagram	39
6	Algorithm Development		40
	6.1	AES Algorithm	41-42
	6.2	Blowfish Algorithm	43
	6.3	DES Algorithm	44
	6.4	Model Development	45-47
	6.5	Result/Output	48-54
7	Conclusion		55
	7.1	Conclusion	56
	7.2	Future Scope	57
	Acknowledgement		58
	References		59

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

Internet is a public-interacted system; the amount of information exchanged over the Internet is completely not safe. Protecting the information transmitted over the network is a difficult task and the data security issues become increasingly important. In recent years, a Controversy has arisen over so-called strong encryption.

This refers to ciphers that are essentially unbreakable without the decryption keys. While most companies and their Customers view it as a means of keeping secrets and minimizing fraud, some governments View strong encryption as a potential vehicle by which terrorists might evade authorities.

These governments, including that of the United States, want to set up a key-escrow Arrangement. This means everyone who uses a cipher would be required to provide the Government with a copy of the key. Decryption keys would be stored in a supposedly secure Place, used only by authorities, and used only if backed up by a court order. Opponents of This scheme argue that criminals could hack into the key-escrow database and illegally obtain, steal, or alter the keys. Supporters claim that while this is a possibility, implementing the key escrow scheme would be better than doing nothing to prevent Criminals from freely using encryption/decryption. At present, various types of Cryptographic algorithms provide high security to information on networks, but they are also have some drawbacks. To improve the strength of these algorithms, we propose a new hybridcryptographic algorithm in this paper. The algorithm is designed using combination of two symmetric cryptographic techniques. These two primitives can be achieved with the Help of Data Encryption Standard (DES) and International Data Encryption Standard (IDEA). This new hybrid cryptographic algorithm has been designed for better security with integrity.

Cloud storage also help in immediate data exchange, thus giving access to multiple people. This makes this service a perfect tool for both distant and in-house work. Thus, online cloud storage and is beneficial for all types of businesses. Cloud storage is a more cost-efficient platform that does not require a huge investment and it can be actively used for connecting and collaborating with clients and employees. Hence more and more users are turning to cloud storage, making it a very popular alternative to traditional storage options.

1.1 Problem Statement:-

As we have discussed in the various issues section that DES is no more secure for transmitting data over the network. It is possible to break the key of DES algorithm with present high performance systems. With 600 million instructions per second we can break the DES within 8 hours. Further if we consider that in future the speed of computer will enhance so it will be possible to break the IDEA algorithm also. So here we are proposing a new hybrid algorithm that is a combination of DES and IDEA. So this hybrid system would have combined security of both the algorithms.

The problem statement of secure file storage on the cloud using hybrid cryptography is how to securely store files on cloud servers while ensuring their confidentiality, integrity, and availability. While cloud storage offers many benefits, such as scalability and cost-effectiveness, it also poses significant security challenges, including unauthorized access, data breaches, and cyber-attacks. Therefore, the problem statement of secure file storage on the cloud using hybrid cryptography is how to design a secure and efficient storage system that addresses these issues and provides secure file storage on the cloud while ensuring confidentiality, integrity, and availability of the data.

Problems Faced by users in previous file storage using Cryptography:-

- * **Key Management:** The security of cryptographic systems relies on the secure storage and distribution of keys used for encryption and decryption. Key management can be a complex issue, especially for large-scale systems or when multiple parties are involved. Key distribution and key revocation are also significant challenges that must be addressed.
- * **Performance:** Cryptographic algorithms, especially those used for public key cryptography, can be computationally intensive and may slow down the storage and retrieval of files. This can be a particular issue when dealing with large files or high volumes of data.
- * **Complexity:** Cryptographic systems can be complex and require specialized knowledge to implement and manage. This complexity can lead to errors, vulnerabilities, and mistakes that can compromise the security of the system.
- * **Compatibility:** Different cryptographic systems may not be compatible with each other, making it challenging to share files securely between different systems or platforms.
- * **Attack and Security:** Cryptographic systems can be vulnerable to various types of attacks, including brute force attacks, side-channel attacks, and cryptographic attacks, such as birthday attacks and chosen plaintext attacks. Additionally, the security of cryptographic systems can be compromised by key management issues, implementation errors, and other vulnerabilities.

Overall, file storage using cryptography requires careful consideration of key management, performance, complexity, compatibility, and security issues to ensure secure and efficient storage and retrieval of files. Hybrid cryptography is one solution that can address some of these challenges by combining the strengths of symmetric and public key cryptography.

1.2 Objectives:-

- 1) To eliminate the need for carrying physical storage devices.
 - 2) To provide Cloud storage safe backup, as opposed to physical storage devices where loss of device, Data.
 - 3) Corruption by a computer virus, natural disasters, amongst other causes, can lead to loss of data.
 - 4) To make the storage more cost-effective & to eliminate the need to invest in hardware.
 - 5) To make storage help developers collaborate and share their work in a more efficient and speedy Manner.
-
- i. **Confidentiality:** To ensure that the files stored on the cloud are protected from unauthorized access and that only authorized users can access them. Hybrid cryptography can provide strong encryption to ensure that the files are kept confidential and secure.
 - ii. **Integrity:** To ensure that the files stored on the cloud are not tampered with or modified by unauthorized users. Hybrid cryptography can provide message integrity checks, such as digital signatures, to verify the authenticity of the data.
 - iii. **Availability:** To ensure that the files stored on the cloud are always available to authorized users, and that the system is designed to handle high volumes of data and user requests.
 - iv. **Key Management:** To securely manage the keys used for encryption and decryption, including key generation, key distribution, and key revocation. Hybrid cryptography can provide a secure and efficient key management system to ensure that the keys are not compromised and that only authorized users have access to them.
-
- i. **Performance:** To ensure that the encryption and decryption process is efficient and does not slow down the storage and retrieval of files. Hybrid cryptography can provide a balance between security and performance by using both symmetric and public key cryptography.
 - ii. **Compliance:** To ensure that the system meets regulatory and compliance requirements, such as GDPR, HIPAA, and PCI DSS.

Overall, the objectives of secure file storage on the cloud using hybrid cryptography are to provide a secure, efficient, and compliant storage system that ensures the confidentiality, integrity, and availability of the data while addressing key management, performance, and compliance issues.

1.3 Scope:-

- i. It eliminates the need for carrying physical storage devices.
 - ii. Cloud storage provides safe backup, as opposed to physical storage devices where loss of device, data corruption by a computer virus, natural disasters, amongst other causes, can lead to loss of data.
 - iii. Cloud storage is more cost-effective as it eliminates the need to invest in hardware.
 - iv. Cloud storage also helps developers collaborate and share their work in a more efficient and speedy Manner.
- ✓ **Data Encryption:** Hybrid cryptography can be used for encrypting data in transit and at rest, ensuring confidentiality and integrity.
 - ✓ **Key Management:** Hybrid cryptography provides a framework for secure key generation, distribution, and management.
 - ✓ **Authentication and Digital Signatures:** Hybrid cryptography can provide authentication and digital signature mechanisms for verifying the authenticity and integrity of data.
 - ✓ **Public Key Infrastructure:** Hybrid cryptography can be used in a public key infrastructure (PKI) to provide secure communication and authentication between parties.
 - ✓ **Cloud Security:** Hybrid cryptography can be used for secure data storage, access control, and identity management in cloud environments.
 - ✓ **Mobile Security:** Hybrid cryptography can be used for securing data in mobile devices, including encryption and authentication.
 - ✓ **Block chain Security:** Hybrid cryptography can be used for secure transactions and data storage in blockchain systems.
 - ✓ **Compliance:** Hybrid cryptography can be used to ensure compliance with regulatory requirements, such as GDPR, HIPAA, and PCI DSS.

Overall, the scope of hybrid cryptography is wide-ranging, and it can be applied to various aspects of information security to provide secure communication, data storage, and access control.

Another advantage of cloud storage could be additional security. The proposed system aims to make the cloud storage System secure using data encryption. Thus, the aim of the proposed system is to increase security of data uploaded onto the cloud by using encryption algorithms to make the system more secure.

1.4 Report Organization

- **In Chapter 2**, we will see the literature survey which will tell us more about the background of the project including the work that has already been done in this field.
- **In Chapter 3**, Planning and Formulation of the project is given. Usage of Agile model and how we integrated and worked around the model.
- **In Chapter 4**, shines light upon the Requirements that are needed and analysis of the system to uncover the additional requirements of the project.
- **In Chapter 5**, the system proposed is introduced which will tell the deep specification of the project and will tell how the different modules of the system will work, the flow of the project regarding data flow, control flow and other flow of the system.
- **In Chapter 6**, we see the implementation of the algorithm of the project and process of model building.
- **In Chapter 7**, Conclusion and Future Scope of this project is mentioned.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 Research Paper Analysis:-

Hybrid Cryptography concept is used for securing storage system of cloud. Two different approaches are used to show the difference between less secure and more secure systems. The first approach uses RSA and AES algorithms; RSA is used for key encryption and AES is used for text or data encryption. In the second or we can say more secured approach, AES and Blowfish algorithms are used. In this approach, these two Algorithms provide double encryption over data and key which provides high security compared to the first one.

To make the centralized cloud storage secure ECC (Elliptic Curve Cryptography) algorithm is implemented. This approach uses single key for encryption and decryption and complete process takes place at the client side. This methodology performs steps such as:

- a. Authentication
- b. Key generation operation
- c. Encryption
- d. Decryption

Modern cryptography originates in the works of Feistel at IBM during the late 1960,s and early 1970,,s. DES was adopted by the NIST, for encrypting unclassified information in 1977. DES is now replaced by the Advanced Encryption Standard (AES), which is a new standard adopted. Another milestone happened during 1978, marked by the publication of RSA. The RSA is the first full-fledged public-key algorithm. This discovery by and large solved the key exchange problem of cryptography. RSA also proposed the world wide acceptable standard techniques like authentication and electronic signatures in modern cryptography.

There are various issues related to DES and IDEA. Some of them are as follows:

- a. The 56-bit key size is the biggest defect of DES. Chips to perform one million of DES encrypt or decrypt operations a second are available (in 1993). A \$1 million DES cracking machine can search the entire key space in about 7 hours.
- b. Hardware implementations of DES are very fast; DES was not designed for software and hence runs relatively slowly.
- c. Brute force is a known-plaintext attack and requires testing, on average, 255 keys.
- d. Differential cryptanalysis is a chosen plaintext attack where the attacker encrypts two chosen plaintext blocks and uses the differences between the cipher text to deduce the key. This attack requires 243 plaintext/cipher text pairs and 255.1 encryption operations, making it less efficient than a brute force attack. Apparently DES was designed to be resistant to differential cryptanalysis.

2.1 Methodology:-

2.1.1 Visual Studio Code (VS Code)

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE. At its heart, Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more .

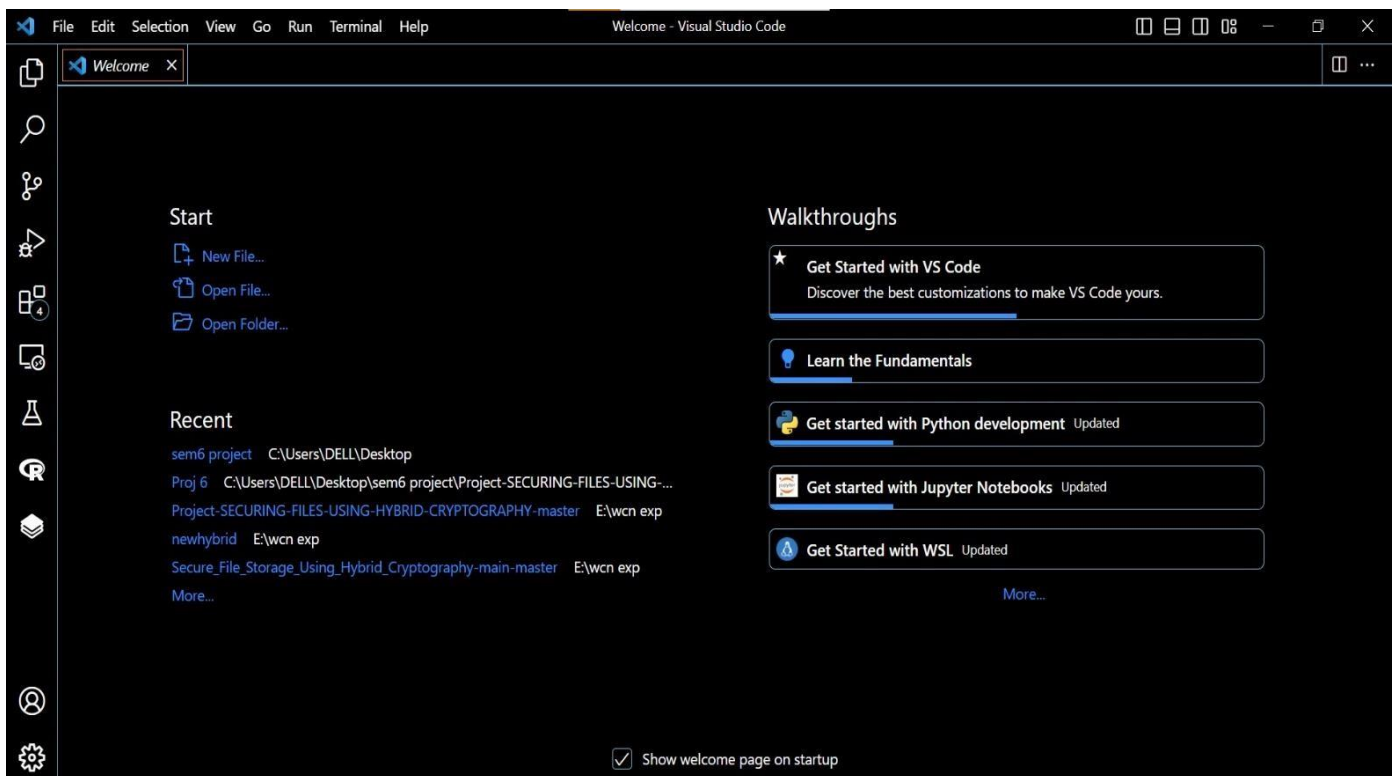
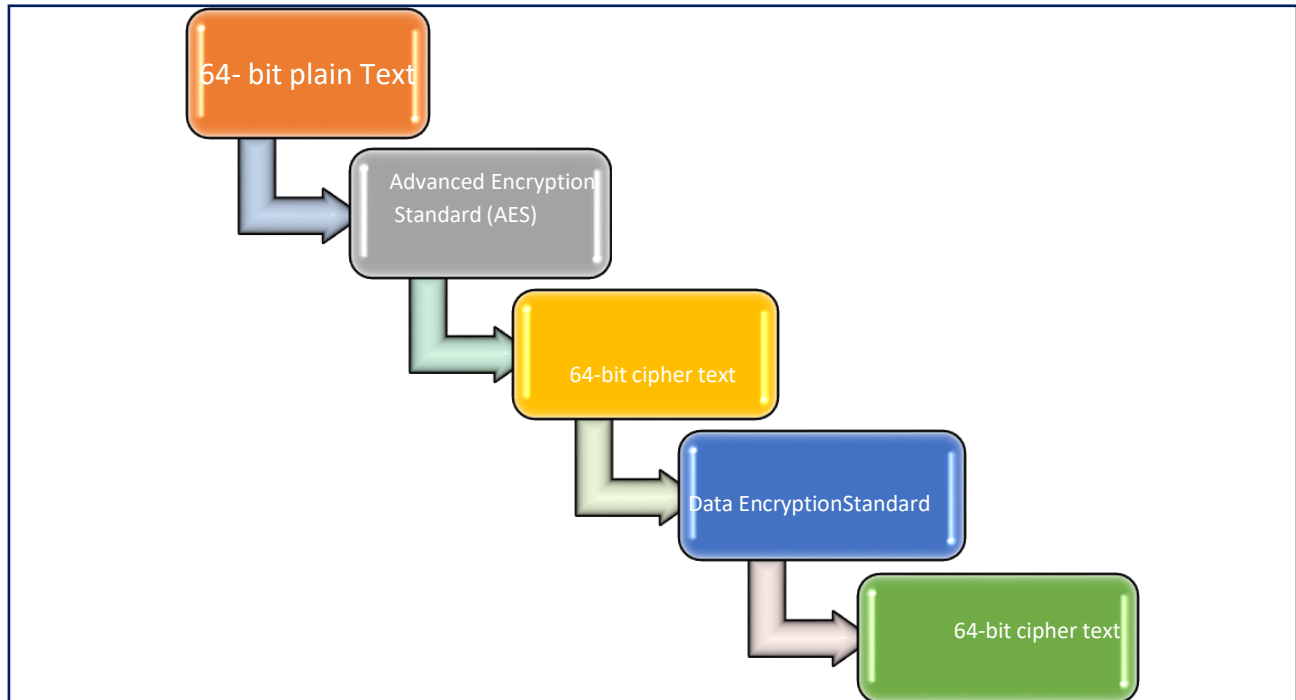


Fig 2.2.1 Visual Studio

□ Steps for Hybrid Cryptography algorithm



DES takes an input of 64-bit plaintext data block and 56-bit key (with 8 bits of parity) and Outputs a 64-bit cipher text block.

1. The 8 parity bits are removed from the key by subjecting the key to its Key Permutation.
2. The plaintext and key are processed in 16 rounds consisting of:
3. The key is split into two 28-bit halves.

Each half of the key is shifted (rotated) by one or two bits, depending on the round. The halves are recombined and subject to a Compression Permutation to reduce the key from 56 bits to 48 bits. This Compressed Key is used to encrypt this round's plaintext block.

- i. The rotated key halves from step 2 are used in next round.
- ii. The data block is split into two 32-bit halves.
- iii. One half is subject to an Expansion Permutation to increase its size to 48 bits.
- iv. Output of step 6 is exclusive-OR'ed with the 48-bit compressed key from step 3.
- v. Output of step 7 is fed into an S-box, which substitutes key bits and reduces the 48-bit

- vi. block back down to 32-bits.
- vii. Output of step 8 is subject to a P-box to permute (scramble) the bits.
- viii. The output from the P-box is exclusive-OR'ed with the other half of the data block.
- ix . The two data halves are swapped and become the next round's input.
- x. After 16 rounds, the resultant is cipher text.
- xi. This resultant cipher text is a input for the IDEA

The 52 16-bit key sub-blocks which are generated from the 128-bit key are produced
As follows:

- i. First, the 128-bit key is partitioned into eight 16-bit sub-blocks which are then directly
Used as the first eight key sub-blocks.
- ii. The 128-bit key is then cyclically shifted to the left by 25 positions, after which the
Resulting128-bit block is again partitioned into eight 16-bit sub-blocks to be directly used
as The next eight key sub-blocks.

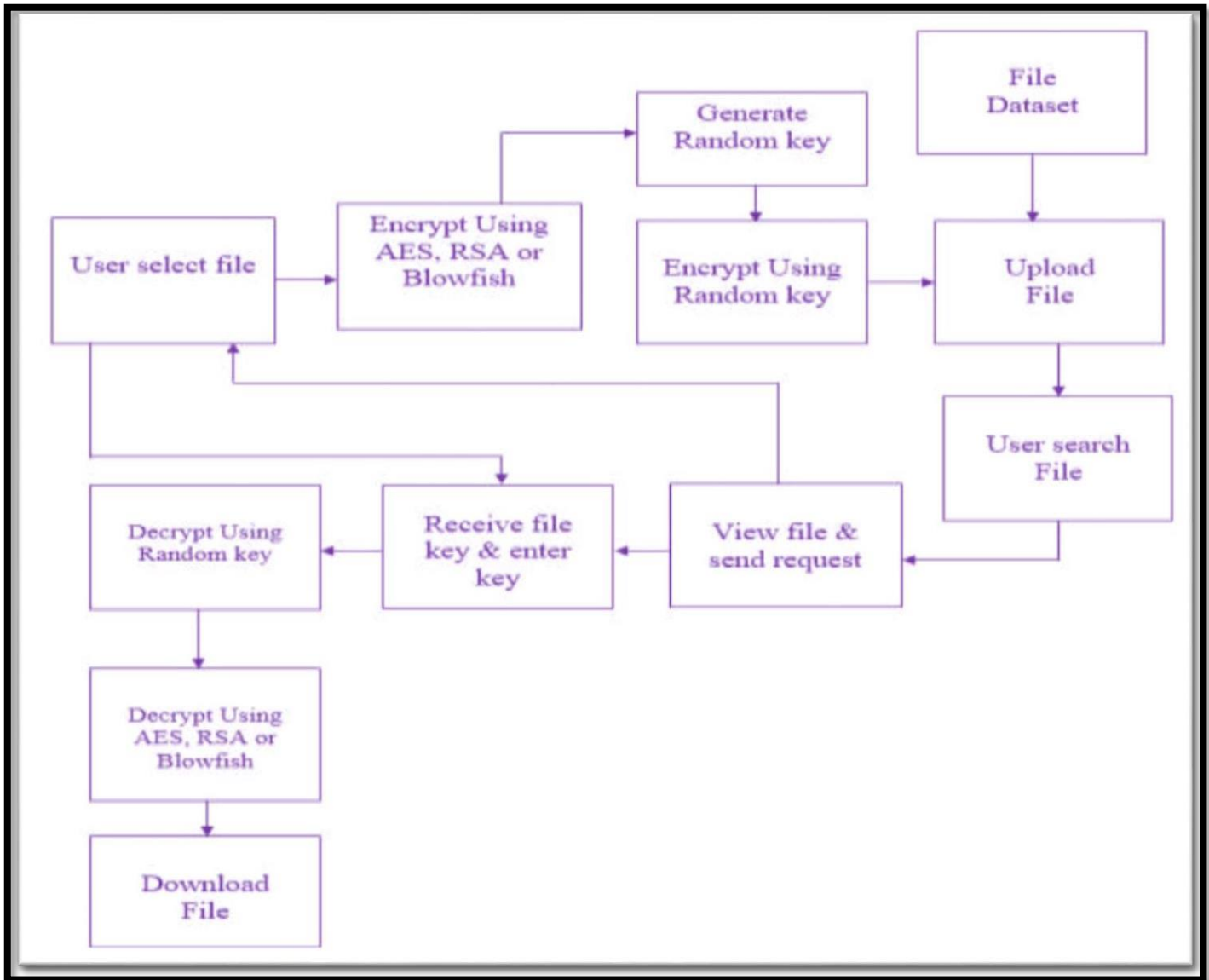
The cyclic shift procedure described above is repeated until all of the required 52 16-bit
Key sub-blocks have been generated

CHAPTER 3

PLANNING & FORMULATION

3. Planning and Formulation

3.1 Project Development Model



Block Diagram Showing Working Of System

3.1.1 Workflow of System:-

The system is designed such that it works in the following way:

- ❖ The user then selects the file that is to be uploaded by browsing from localstorage.
- ❖ The user then selects the encryption algorithm that they want to use. The proposed system provides the choice between using a combination of AES and RSA or AES and Blowfish.
- ❖ The selected file gets uploaded after getting encrypted using the selected encryption algorithm combination.
- ❖ The user also has the option of viewing the files that they have uploaded or have access to and downloading them.

3.1.2 Advantage of Hybrid Cryptography in Model:-

- ✓ Secrecy: Nobody will be able to get any information about the encrypted plaintext (except the length), unless they have access to the secret key.
- ✓ Asymmetry: Encrypting the cipher text can be done with the public key, but for decryption the secret key is required.
- ✓ Cloud cryptography adds a high layer of security and prevents a data breach by encrypting data stored in the cloud.
- ✓ Data used or stored in the cloud is protected using encryption mechanisms. Since all data stored by cloud providers is encrypted, users can access shared cloud services securely.

3.1.3 Disadvantage of Hybrid Cryptography in Model:-

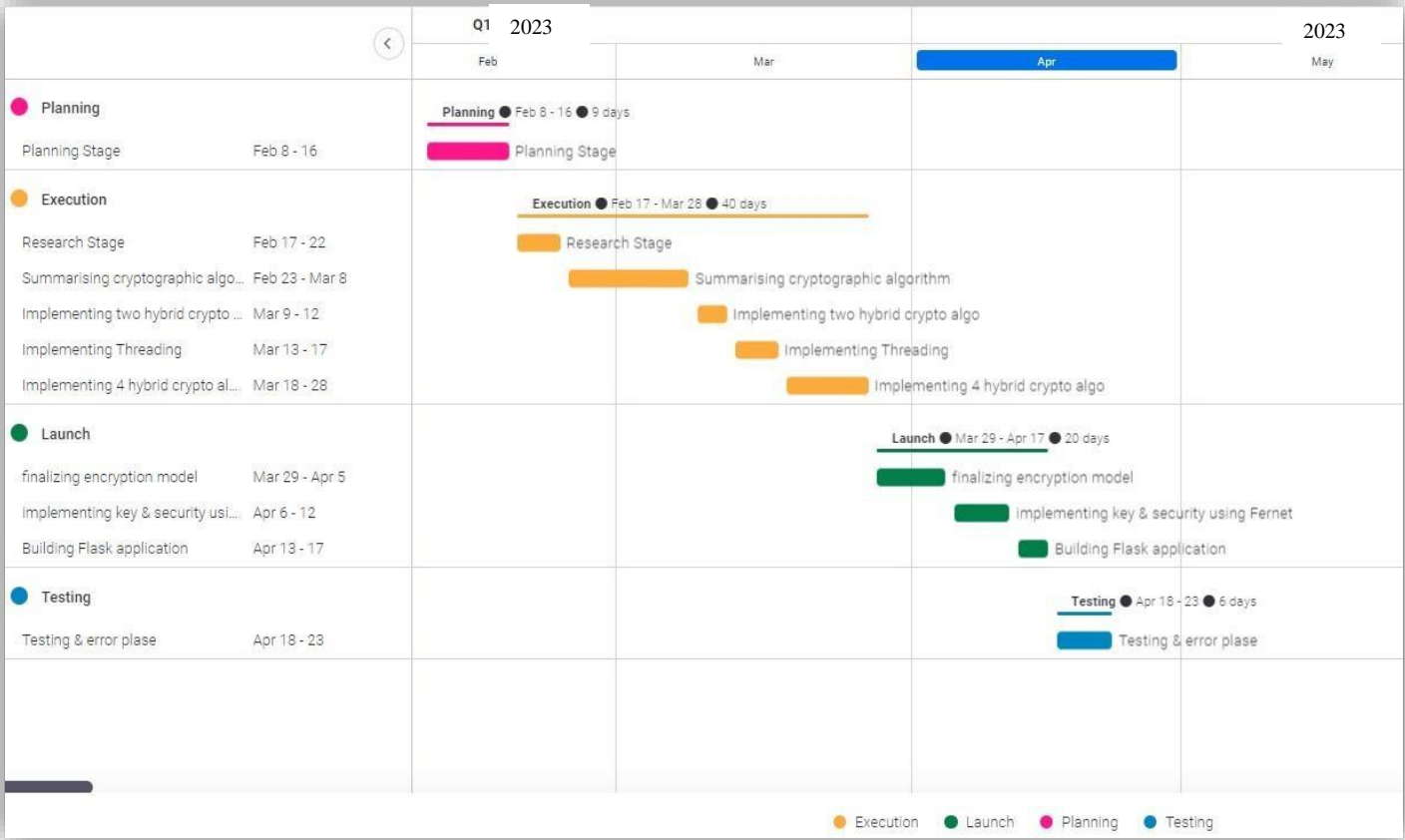
- Difficulty of implementation
- Visibility.
- Compatibility.
- Expense.
- Cost savings.
- Control.
- Scalability and Deployment.

3.2 Timeline Chart

Months	January				February				March				April				May			
Phases	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Requirements Analysis																				
Feasibility Study																				
Design and architecture																				
Product Development																				
Implementation																				
Testing																				
Report Organization																				

Table 3.1 Timeline Chart

3.3 Gantt chart



Gantt Chart

3.3 Feasibility Analysis

Next step in analysis is feasibility study. By performing a feasibility study the scope of the system will be defined completely. Most computer systems are developed to satisfy a known user requirement. This means that the first event in the life cycle of a system is usually the task of studying whether it is feasible to computerize a system under consideration or not. Once the decision is made, a report is forwarded and it is known as Feasibility Report. The feasibility is studied under the following contexts:

3.3.1 Technical Feasibility

It involves determining whether or not a system can actually be constructed to solve the problem at hand. The technical issues raised during the feasibility stage of investigation are related to achievability of project's goal and possibility of completion of project.

3.3.2 Economical Feasibility:

This feasibility deals with the cost/benefit analysis. A number of intangible benefits like user friendliness, robustness and security were pointed out. The cost that will be incurred upon the implementation of this project would be quite nominal.

3.3.1 Operational Feasibility:

The developed system will be very reliable and user friendly. All the features and operations that we will implement in our project are possible to implement and thus feasible. This will facilitate easy use and adoptability of the system. With the use of menus, and proper validation required it becomes fully understandable to the common user and operational with the user.

CHAPTER 4

REQUIREMENTS ANALYSIS

4. Requirements Analysis

4.1 Hardware Requirement

1. Processor: Intel i7 (9th Gen)
2. Hard Disk: 1 TB or more
3. Computer / Laptop
4. Webcam and Microphone
5. RAM: 8 GB or above

4.2 Software Requirement

1. Operating System : Windows 10/11
2. Visual Studio
3. Python
4. Flask
5. HTML
6. CSS
7. JavaScript

4.3 Functional Requirement

1. In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.
2. Python: Support for hybrid cryptography: Python has built-in support for many cryptographic algorithms, including those used in hybrid cryptography. This includes algorithms like RSA, AES, and SHA. Python is a well-suited language for secure cloud storage projects using hybrid cryptography due to its ease of use, availability of libraries, and support for cryptographic algorithms, cross-platform compatibility, and flexibility.
3. HTML – It is a technology which is used to define the structure of any webpage. It is also used to specify whether your content should be in a paragraph, list, heading, link, image, multimedia player, form, or one of many other available elements or even a new element that you define
4. CSS - Cascading Style Sheets is a style sheet language which is used for describing the styling of a document written in a markup language such as HTML. Without CSS, every web page would be drab plain text and images that flowed straight down the page.
5. Bootstrap – It is a free and an open-source tool of collections which is used for creating responsive websites and web applications. It is known for popular uses with HTML, CSS,

And JavaScript framework for developing responsive, mobile and desktop websites. It solves the issues of compatibility. We get Faster and Easier Websites through bootstrap. a) Development. b) It creates Platform-independent web pages.

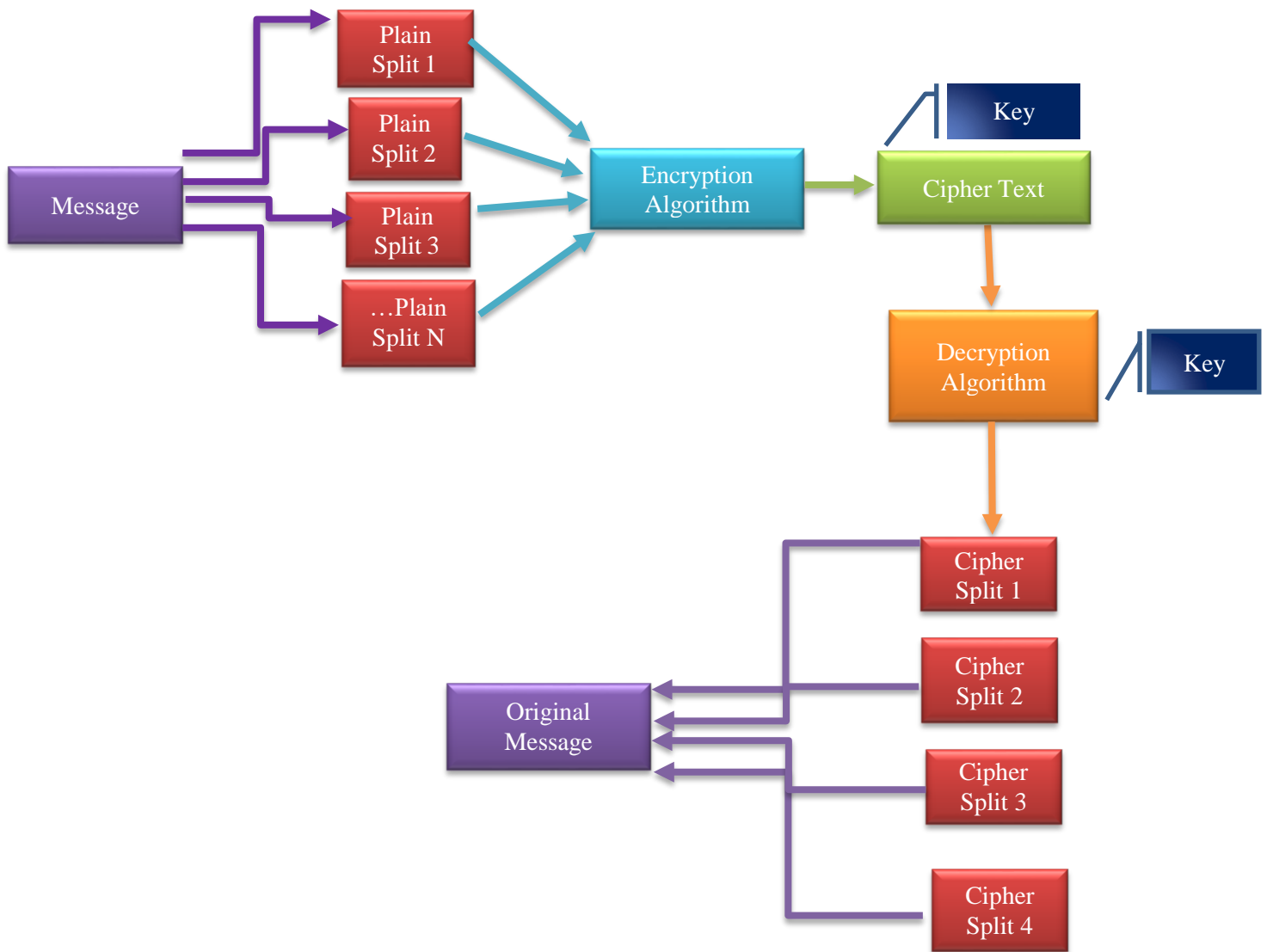
4.4 Non-Functional Requirements:-

1. **Security:** The most important non-functional requirement is the security of the data stored in the cloud. The project should provide strong encryption, data integrity, and access control mechanisms to ensure the confidentiality, integrity, and availability of the data.
2. **Performance:** The project should be designed to handle large volumes of data efficiently, with minimal impact on system performance. This includes fast encryption and decryption times, low latency, and high throughput.
3. **Scalability:** The project should be scalable to support increasing amounts of data and users over time. This includes the ability to add additional storage capacity and processing power as needed.
4. **Reliability:** The project should be highly available and resilient, with built-in redundancy and failover mechanisms to ensure that data is always available.
5. **Compatibility:** The project should be compatible with a range of operating systems, platforms, and devices to ensure that users can access their data from anywhere.
6. **Usability:** The project should be easy to use, with a simple and intuitive user interface that makes it easy for users to store and retrieve their data.
7. **Compliance:** The project should comply with relevant regulatory requirements, such as data privacy and security regulations.
8. **Maintenance:** The project should be easy to maintain and update, with clear documentation and support from the development team.

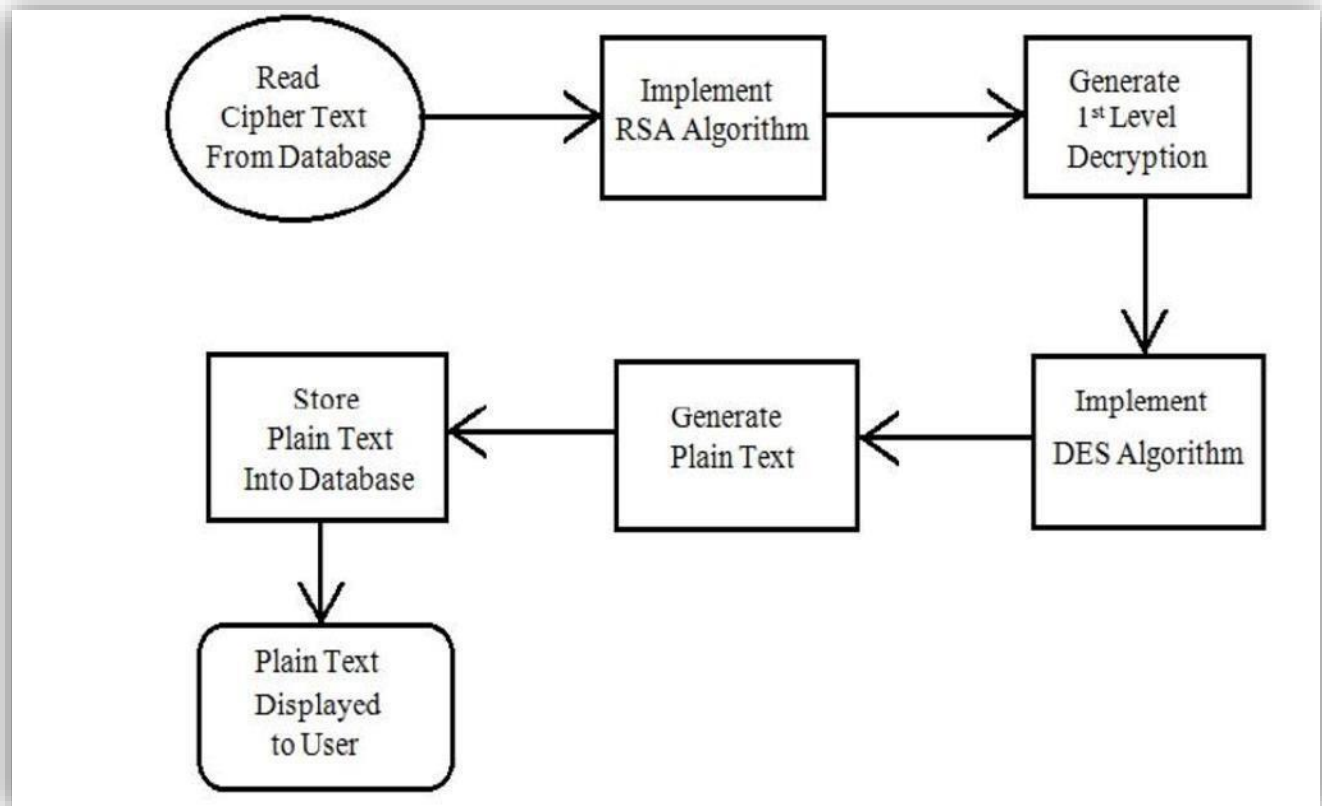
Overall, non-functional requirements for secure cloud storage using hybrid cryptography project are focused on ensuring the security, performance, scalability, reliability, compatibility, usability, compliance, and maintainability of the system.

4.5 Data Flow Diagram

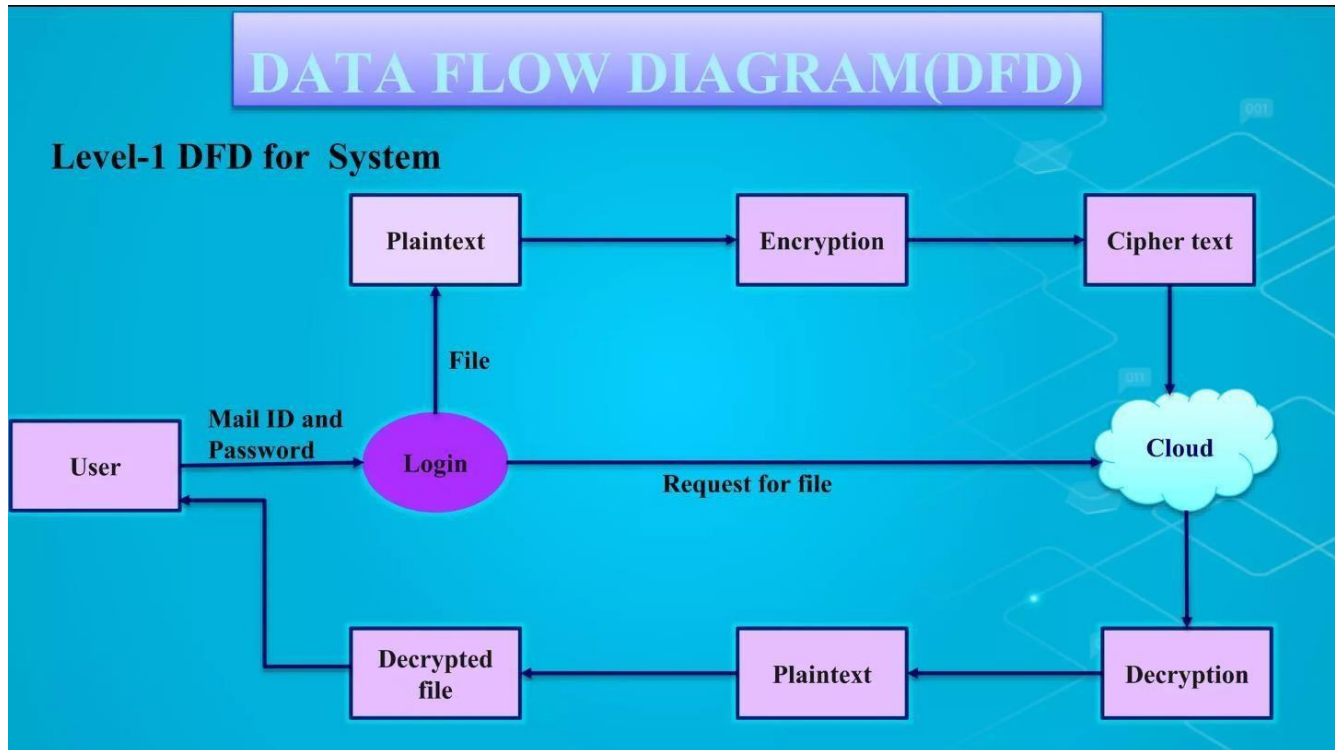
4.5.1 DFD LEVEL 0:-



Dataflow Diagram of multilevel Decryption



4.5.2 DFD LEVEL 1:



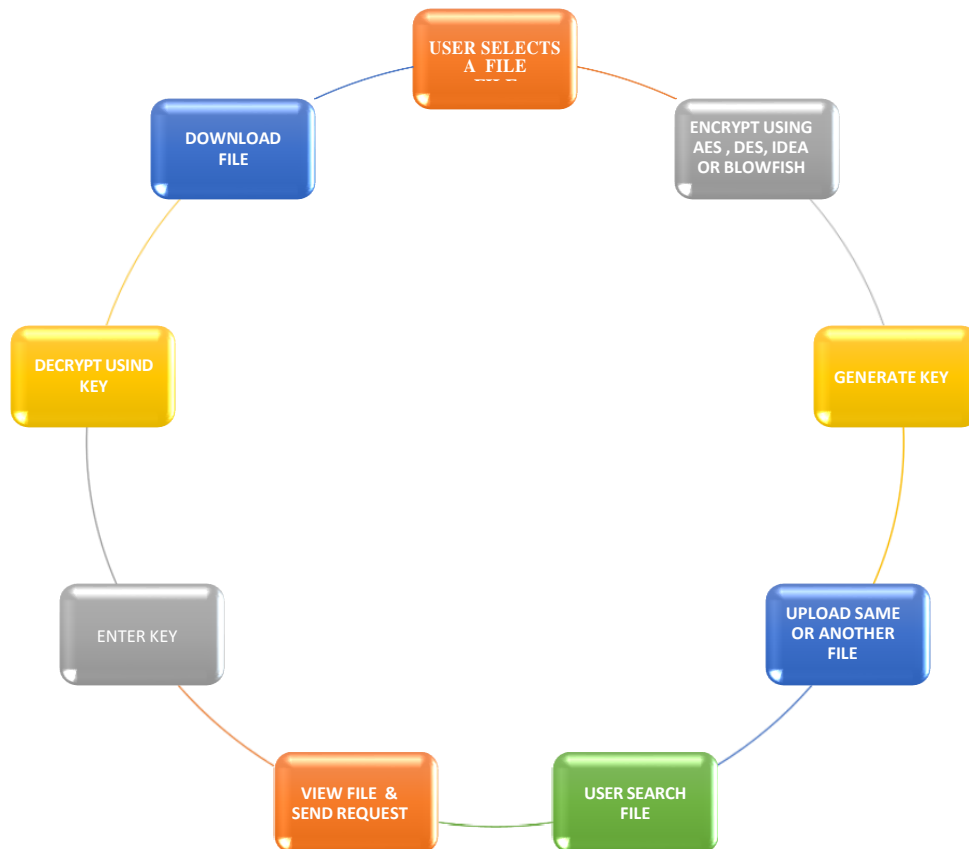
CHAPTER 5

SYSTEM DESIGN

5. System Design

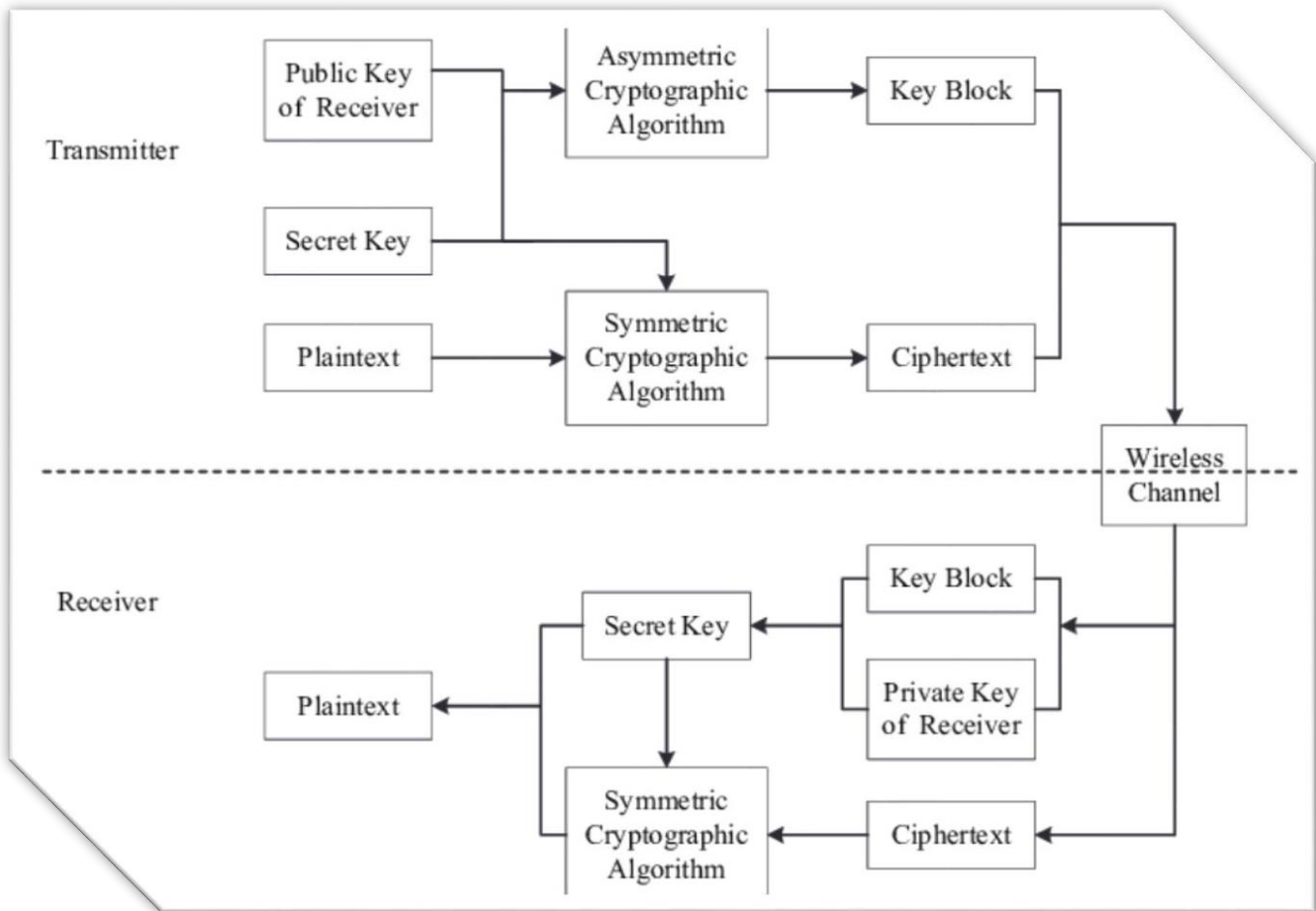
5.1 Flow Chart

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

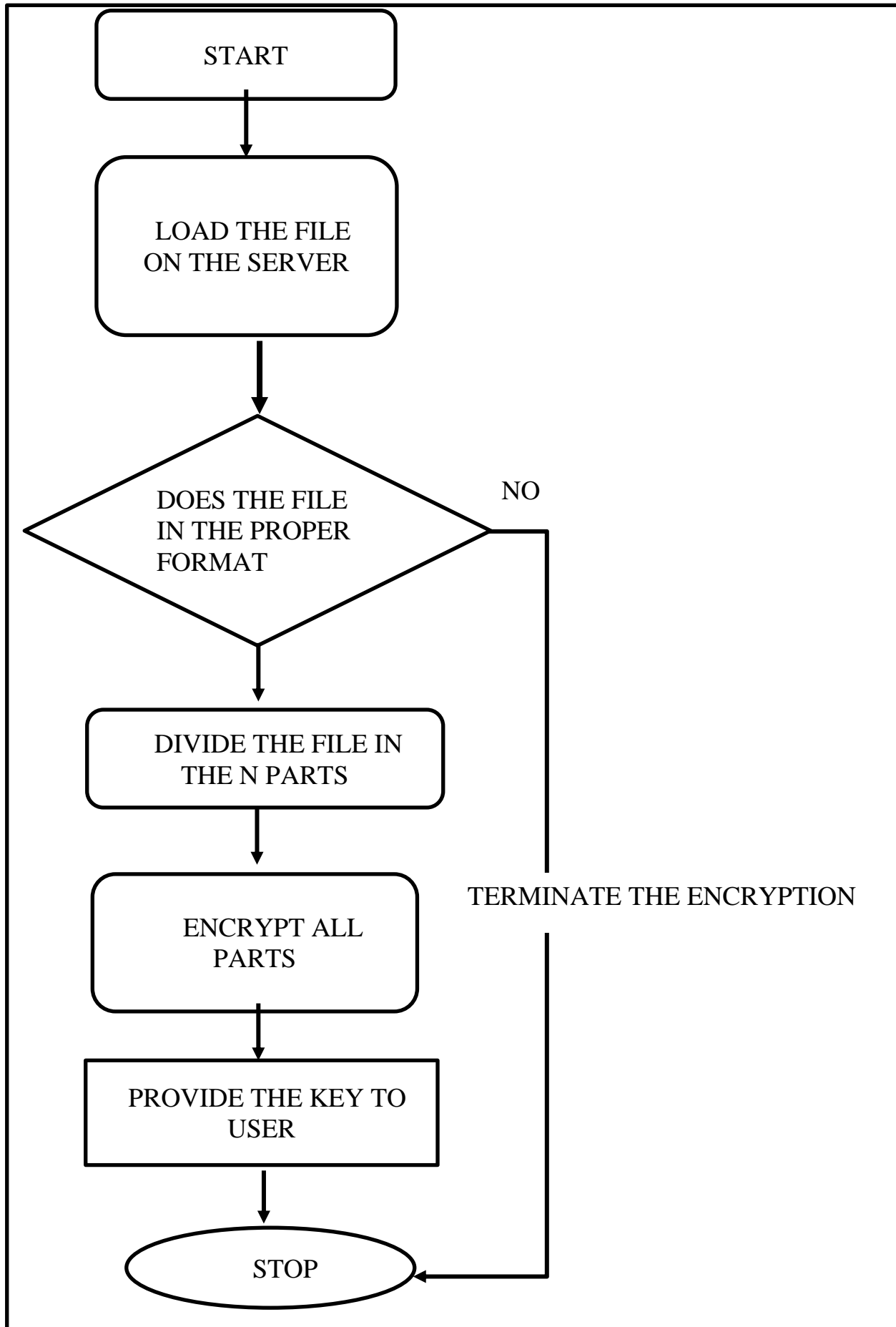


This flow chart represents flow of data or say the users flow while accessing the software.

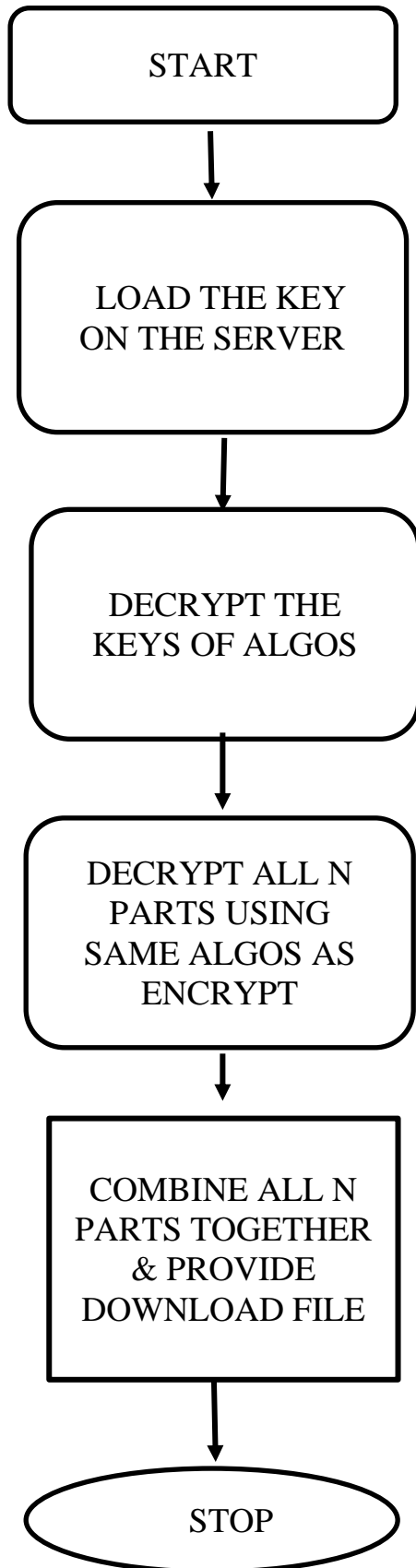
5.2 Flowchart



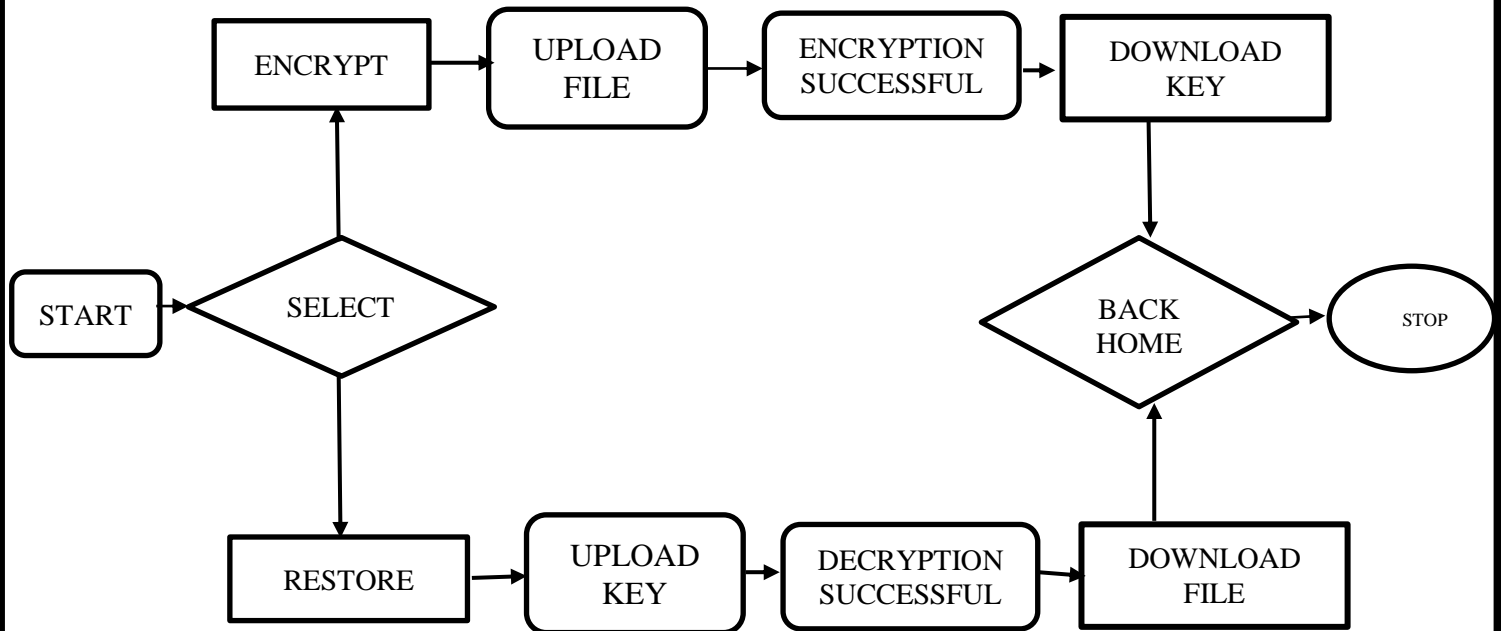
ENCRYPTION FLOWCHART:-



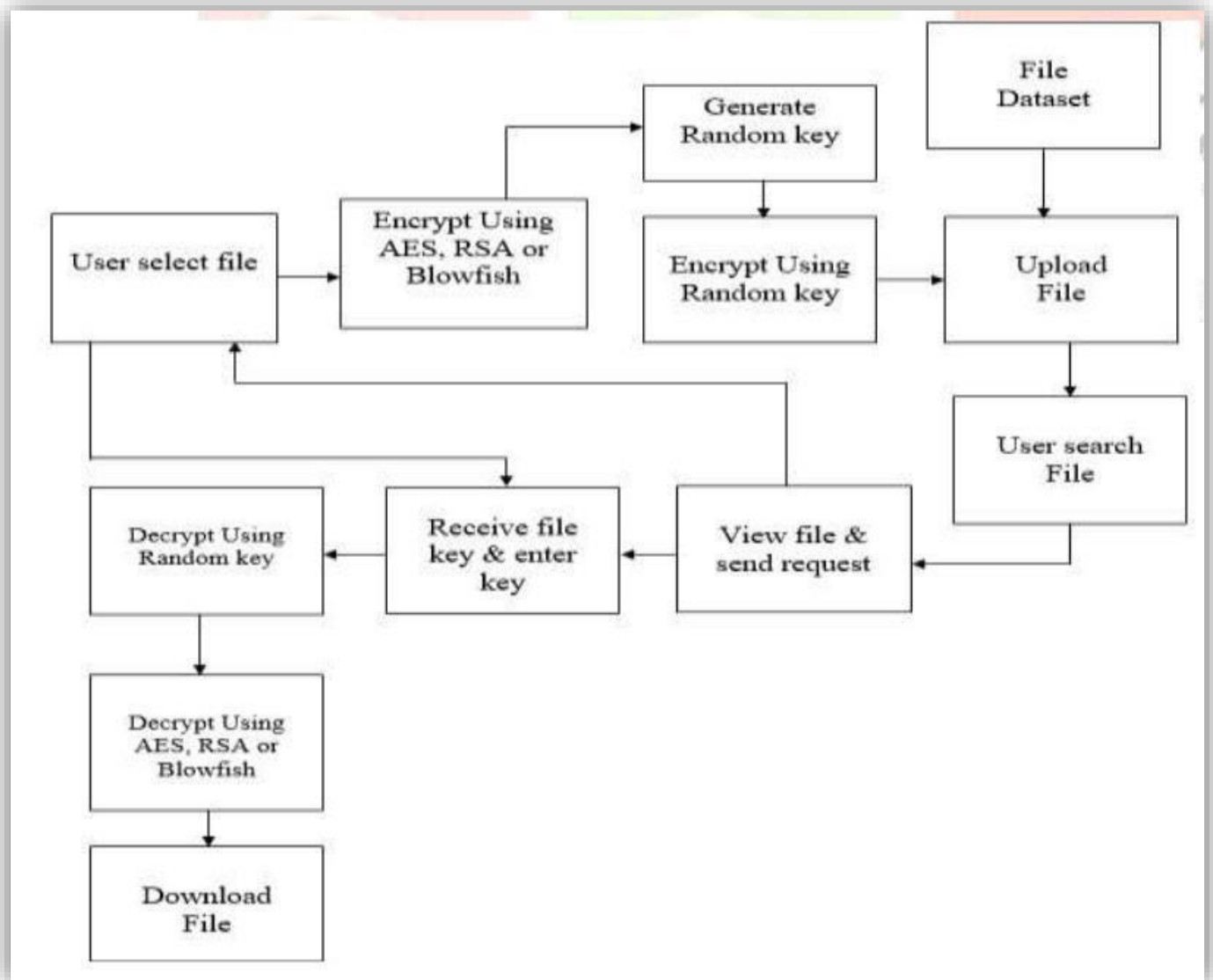
DECRYPTION FLOWCHART:-



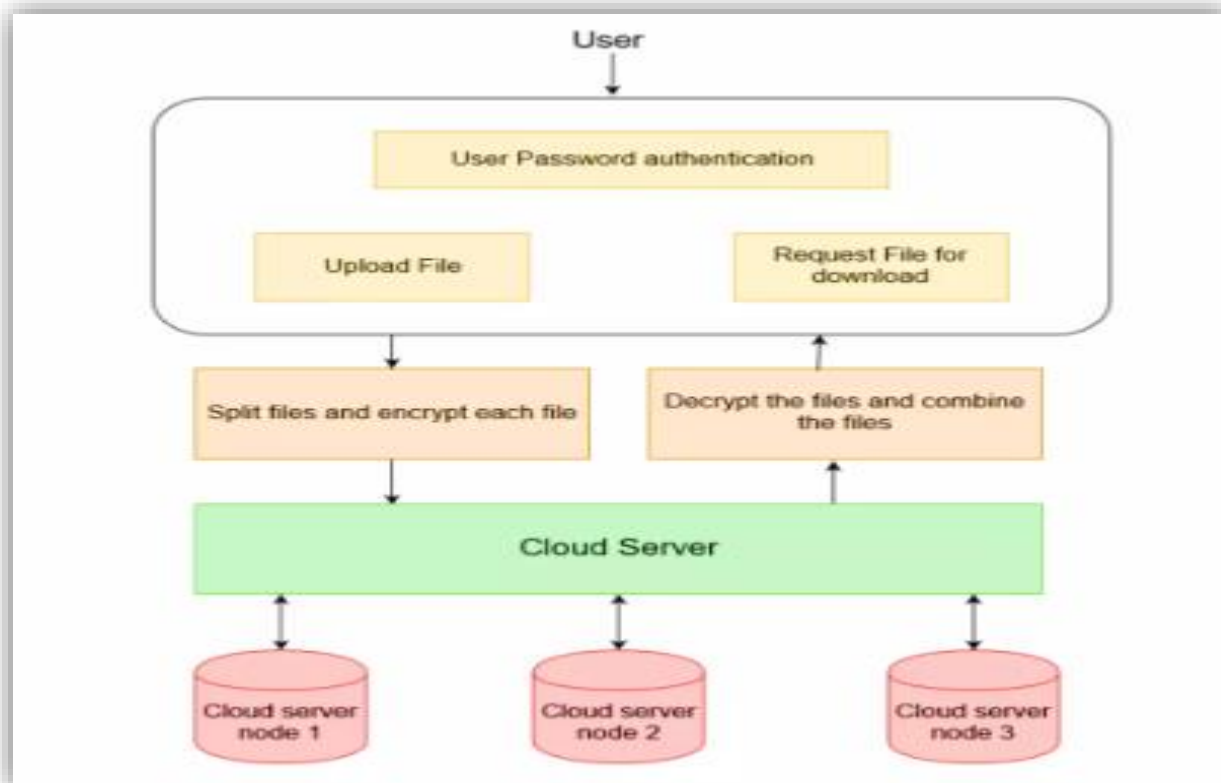
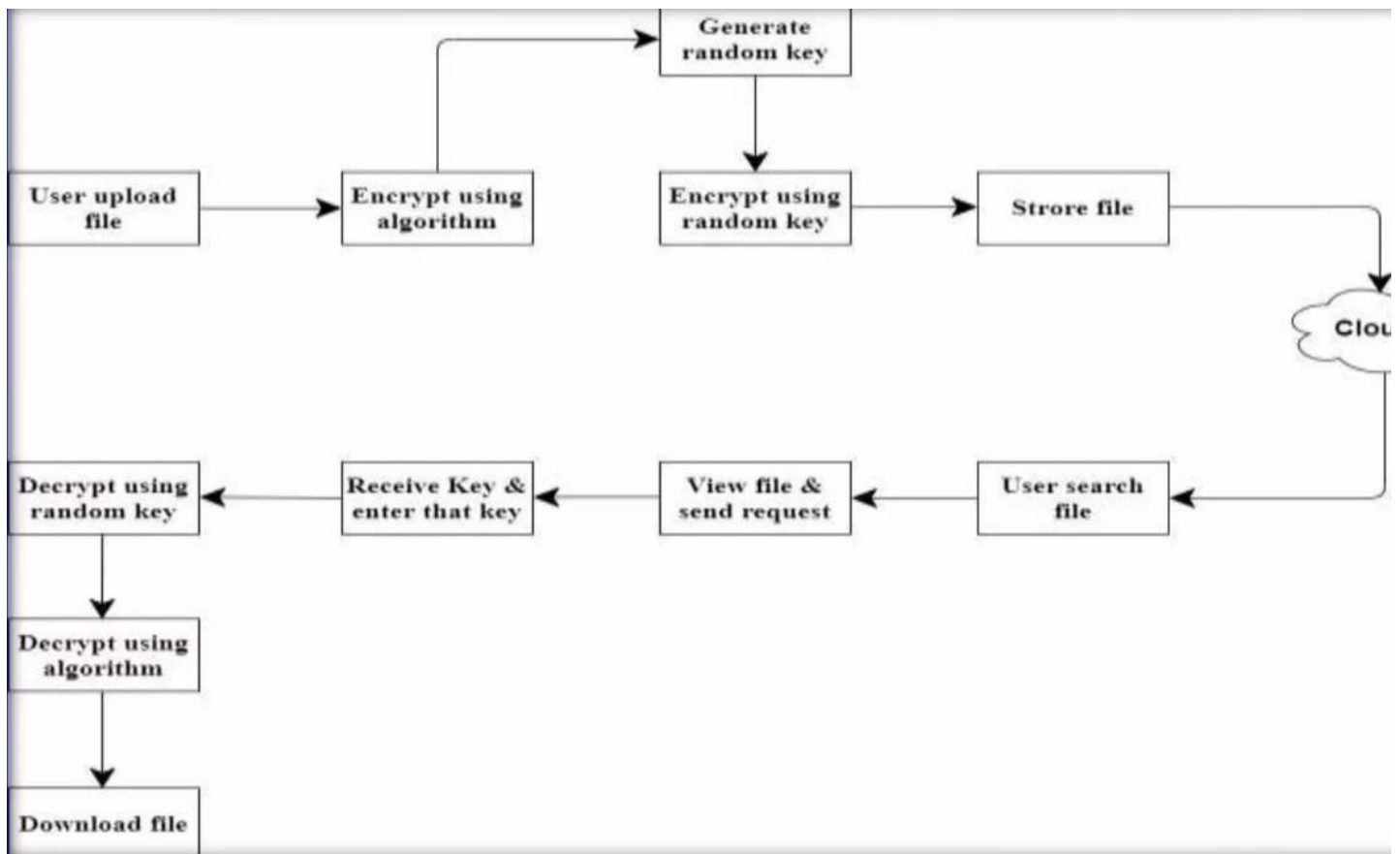
FLOW OF SYSTEM



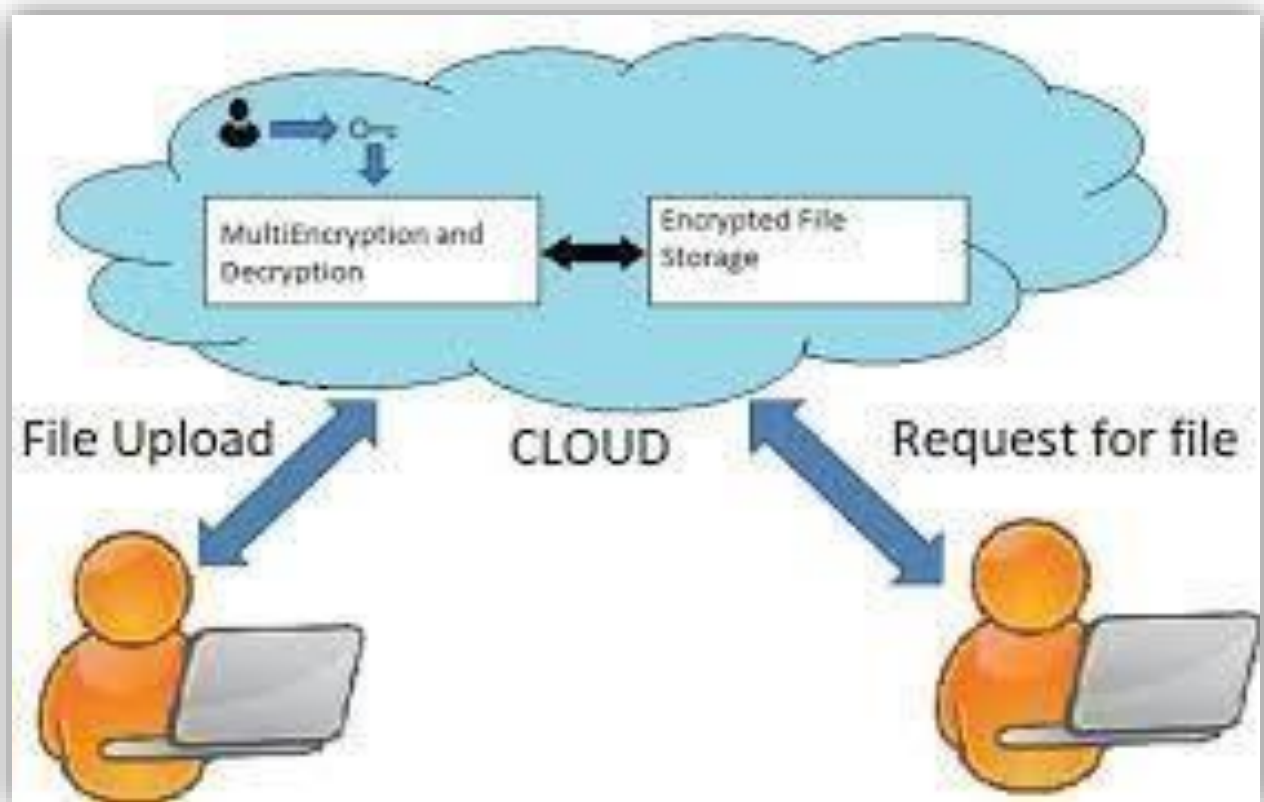
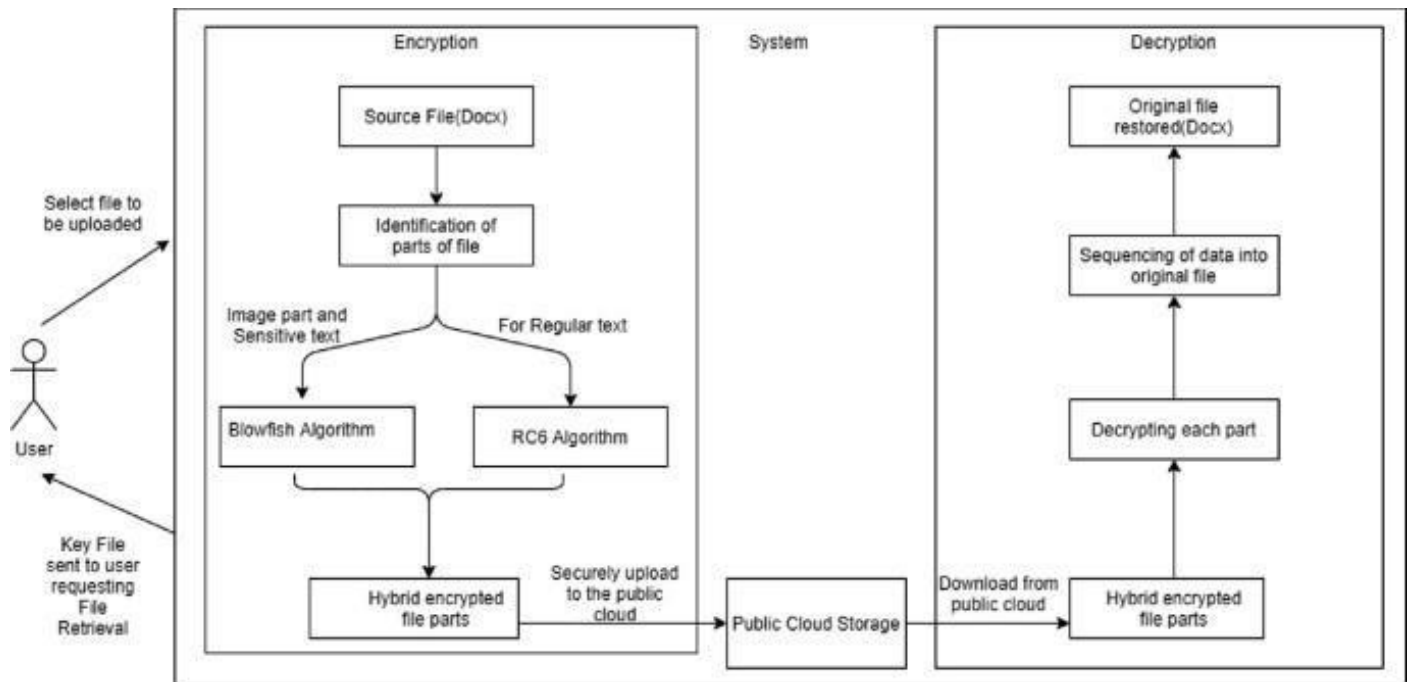
5.3 Sequence Diagram:-



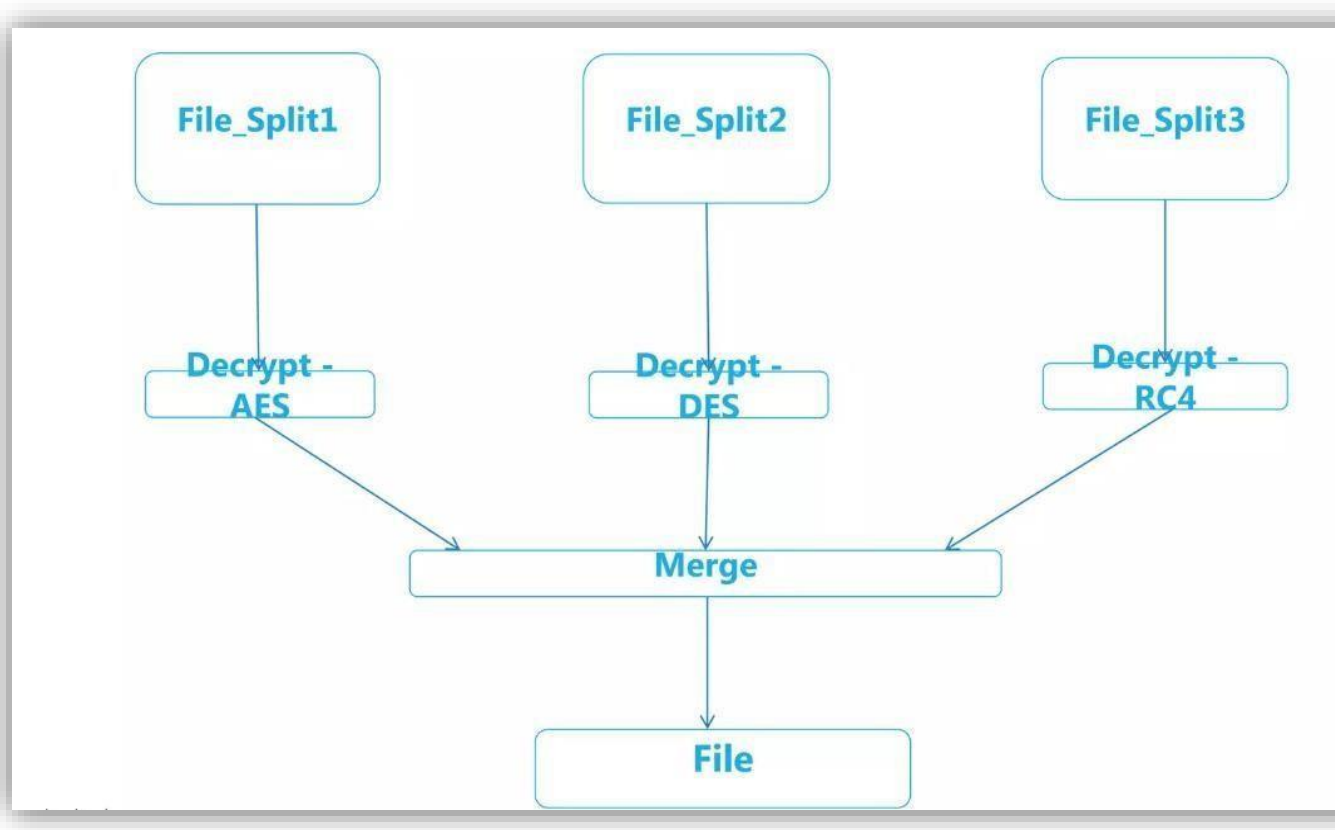
5.4 System Architecture:-



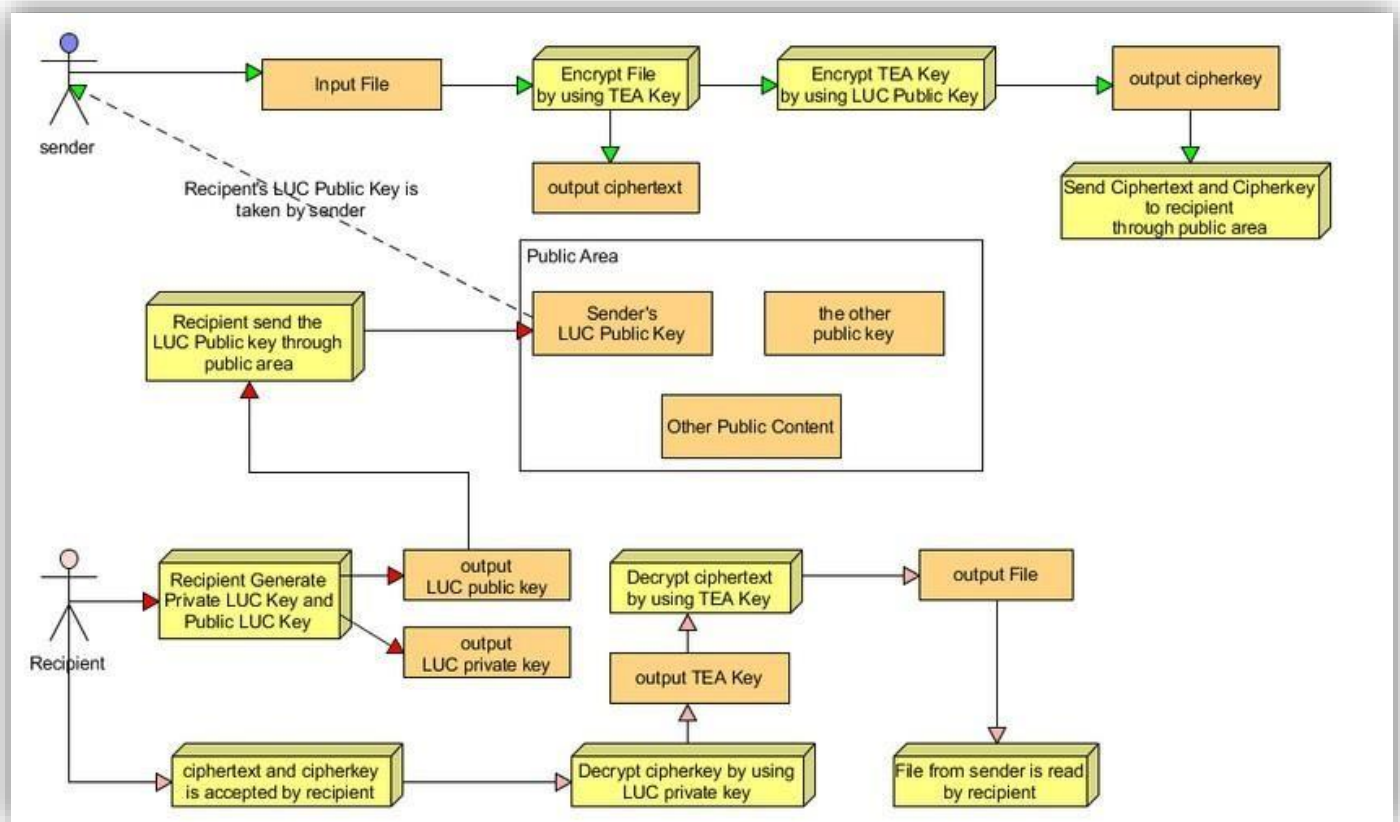
5.5 UML Diagrams



5.6 DECRYPTING & MERGING FILE:-



5.7 ER Diagram:-



CHAPTER 6

ALGORITHM DEVELOPMENT

6. Algorithm Development

6.1 Algorithm Development

AES Algorithm

The Advanced Encryption Standard (AES) also known as 'Rijndael' is a symmetric-key block cipher algorithm having three fixed 128-bit block ciphers with cryptographic key sizes of 128, 192 and 256 bits respectively.

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.

Step-wise description of the algorithm:

Key Expansions:

Round keys are derived from the cipher key using AES key schedule, it also requires a separate 128-bit roundkey block for each round plus one more.

Initial Round:

Add Round Key - using bitwise xor each byte of the state is combined with a block of the round key.

Rounds:

Sub Bytes - according to a lookup table each byte is replaced with another in a non-linear substitution step.

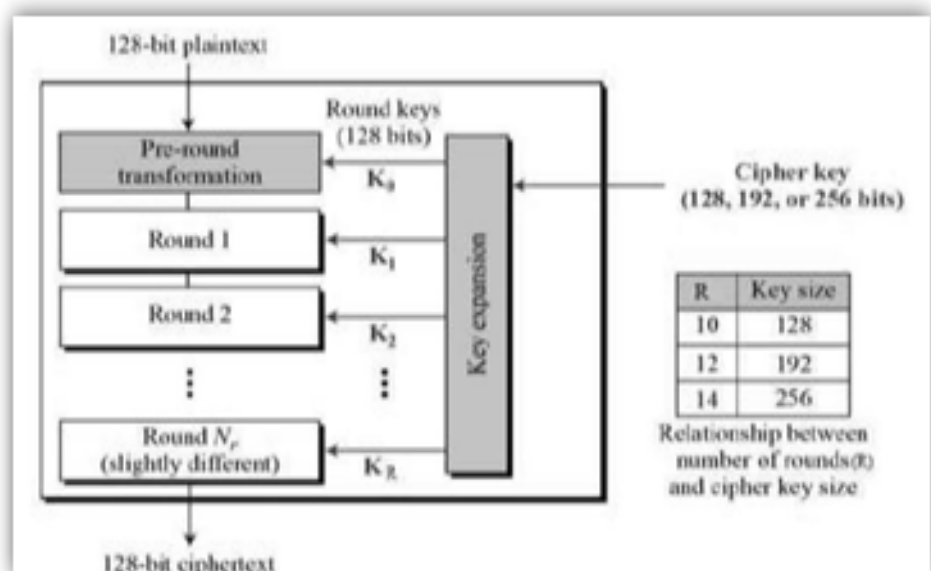
(a) Shift Rows - a transposition step where the last 3 rows of the state are shifted cyclically a certain number of steps.

(b) Mix Columns - a mixing operation which operates on the columns of the state, combining the 4 bytes in each column.

(c) Add Round Key

Final Round (no Mix Columns).

- I. Sub Bytes
- II. Shift Rows
- III. Add Round Key



Blowfish Algorithm

Blowfish is a symmetric block encryption algorithm designed which is fast, compact, simple and secure to use. As follows:

It encrypts data on large 32-bit microprocessors at a rate of 26 clock cycles per byte and can run in less than 5K of memory. It uses addition, XOR, lookup table with 32-bit operands. Also the key length is variable, it can be in the range of 32-448 bits: default 128 bits key length. It is suitable for applications where the key does not change often, like communication link or an automatic file encryptor. It is unpatented and royalty-free.

Description of Algorithm:

Blowfish symmetric block cipher algorithm encrypts block data of 64-bits at a time. It will follow the 16 rounds Feistel network and this algorithm is divided into two parts.

1. Key-expansion
2. Data Encryption

Key-expansion:

It will convert a key into several sub key arrays totalling 4168 bytes consisting at most 448 bits.

Blowfish uses five subkey-arrays:

One 18-entry P-array consisting of 32-bit sub keys:

P1, P2,, P18 and four 256-entry S-boxes of 32-bit each:

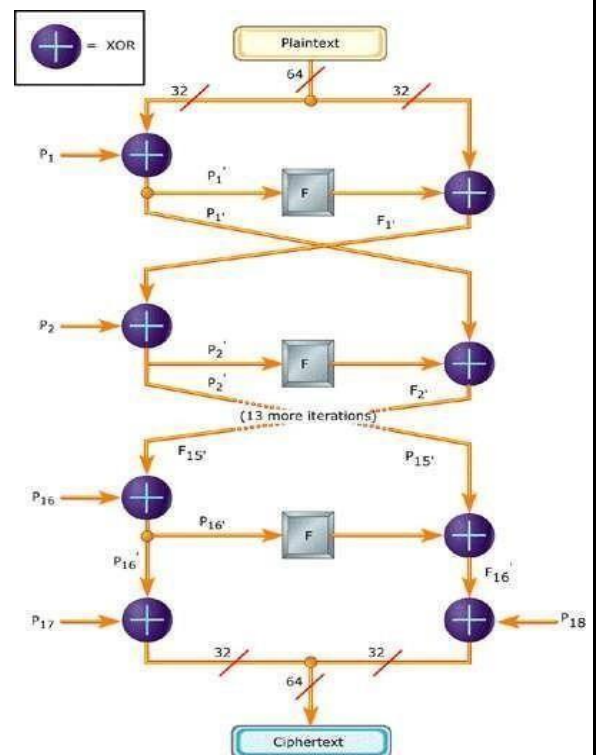
S1,0, S1,1,, S1,255

S2,0, S2,1,, S2,255

S3,0, S3,1,, S3,255

S4,0, S4,1,, S4,255

These keys are generated earlier to any data encryption or decryption.



Working of Blowfish Algorithm

Data Encryption:

It is having a function to iterate 16 times of network. Each round consists of key-dependent permutation and a key and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookup tables for each round.

Algorithm: Blowfish Encryption

Divide x into two 32-bit halves: x_L, x_R For $i = 1$ to

16: $x_L = x_L \text{ XOR } P_i$ $x_R = F(x_L) \text{ XOR } x_R$

Swap x_L and x_R

Swap x_L and x_R (Undo the last swap.) $x_R = x_R \text{ XOR } P_{17}$ $x_L =$

$x_L \text{ XOR } P_{18}$

Recombine x_L and x_R

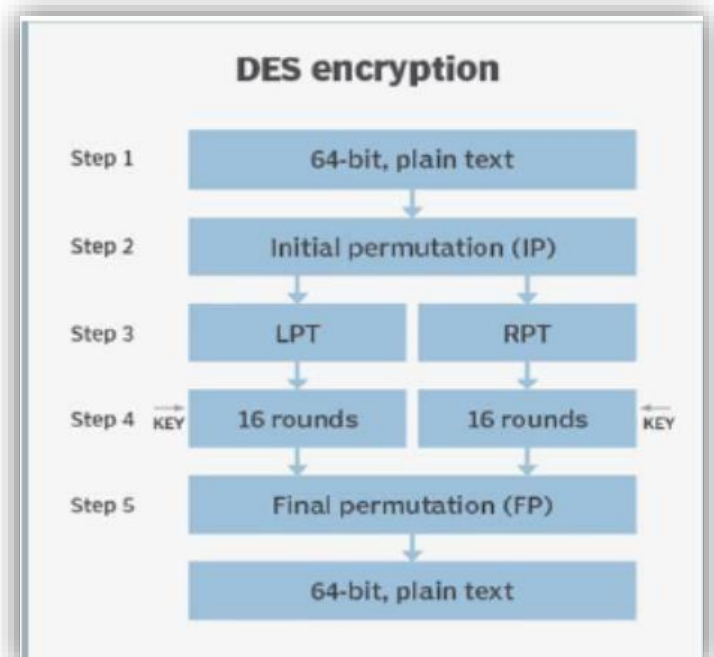
DES Algorithm:-

Data Encryption Standard (DES) is a symmetric-key block cipher.

Encrypts data in blocks of size of 64 bits each and Key length is 56 bits.

DES is based on the two attributes: substitution and transposition.

DES consists of 16 rounds. The result of this process produces 64-bit cipher text.



6.2 Model Development

Implementation of Secure Cloud Storage using Hybrid Cryptography System

```
import os
from flask import Flask, request, redirect, url_for, render_template,
send_from_directory, flash
from werkzeug.utils import secure_filename
from dataProcessing
import *
from Threads
import *
from flask import
send_file import
time
import
os
script
= "
```

```
UPLOAD_FOLDER = ''
ALLOWED_EXTENSIONS =
set(['txt'])
```

```
# api = API(app)
app = Flask(____name____)
app.config['UPLOAD_FOLDER'] =
UPLOAD_FOLDER
app.config["CACHE_TYPE"] = "null"
```

```
def resultE():
    path =
    "./Segments"
    dir_list =
    os.listdir(path)
    print(dir_list)
    return render_template('Result.html',dir_list = dir_list)
```

```
def resultD():
    return render_template('resultD.html')
```

```
@app.route('/encrypt')
def EncryptInput():
    Segment()
    gatherInfo()
```

```

HybridCrypt()
    return resultE()

@app.route('/decrypt')
def DecryptMessage():
    st=time.time()
    HybridDeCrypt()
    et=time.time()
    print(et-st)
    trim()
    st=time.time()
    e() Merge()
    et=time.time()
    e() print(et-st)
    return resultD()
def start():
    content = open('./Original.txt','r')
    content.seek(0)
    first_char = content.read(1)
    if not first_char:
        return render_template('Empty.html')

    else:
        return render_template('Option.html')

@app.route('/')
def index():
    return render_template('index.html')

def allowed_file(filename):
    return '.' in filename and \
        filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

@app.route('/return-files-key')
def return_files_key():
    try:
        return send_file('./Original.txt',attachment_filename='Original.txt',as_attachment=True)
    except:
        return render_template('Error.html')

```



```
hment=True) except Exception as e:
    return str(e)
```

```
@app.route('/return-files-
data/')
def
return_files_data():
    try:
        return
    send_file('./Output.txt',attachment_filename='Output.txt',as_attach
ment=True) except Exception as e:
        return str(e)
```

```
@app.route('/data/',
methods=['GET', 'POST'])
def
upload_file():
    if request.method == 'POST':
        if 'file' not in request.files:
            return
        render_template('Nofile.htm
l')file = request.files['file']
        if file.filename == "":
            return
        render_template('Nofile.htm
l')
        if file and
        allowed_file(file.filename):
            filename = secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'],
'Original.txt'))return start()

    return render_template('Invalid.html')
```

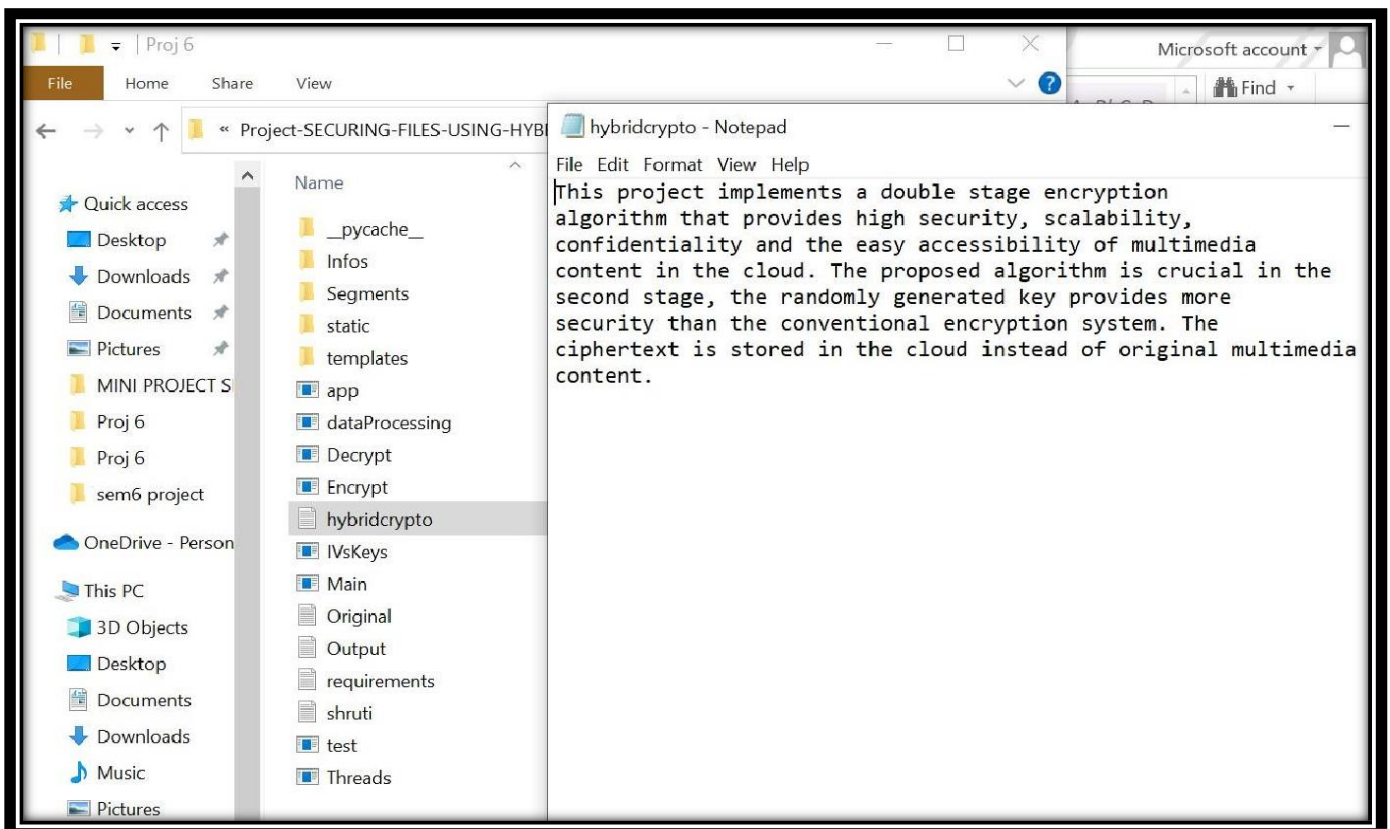
```
if __name__ == '__main__':
    app.run(debug=True)
```

OUTPUTS:-

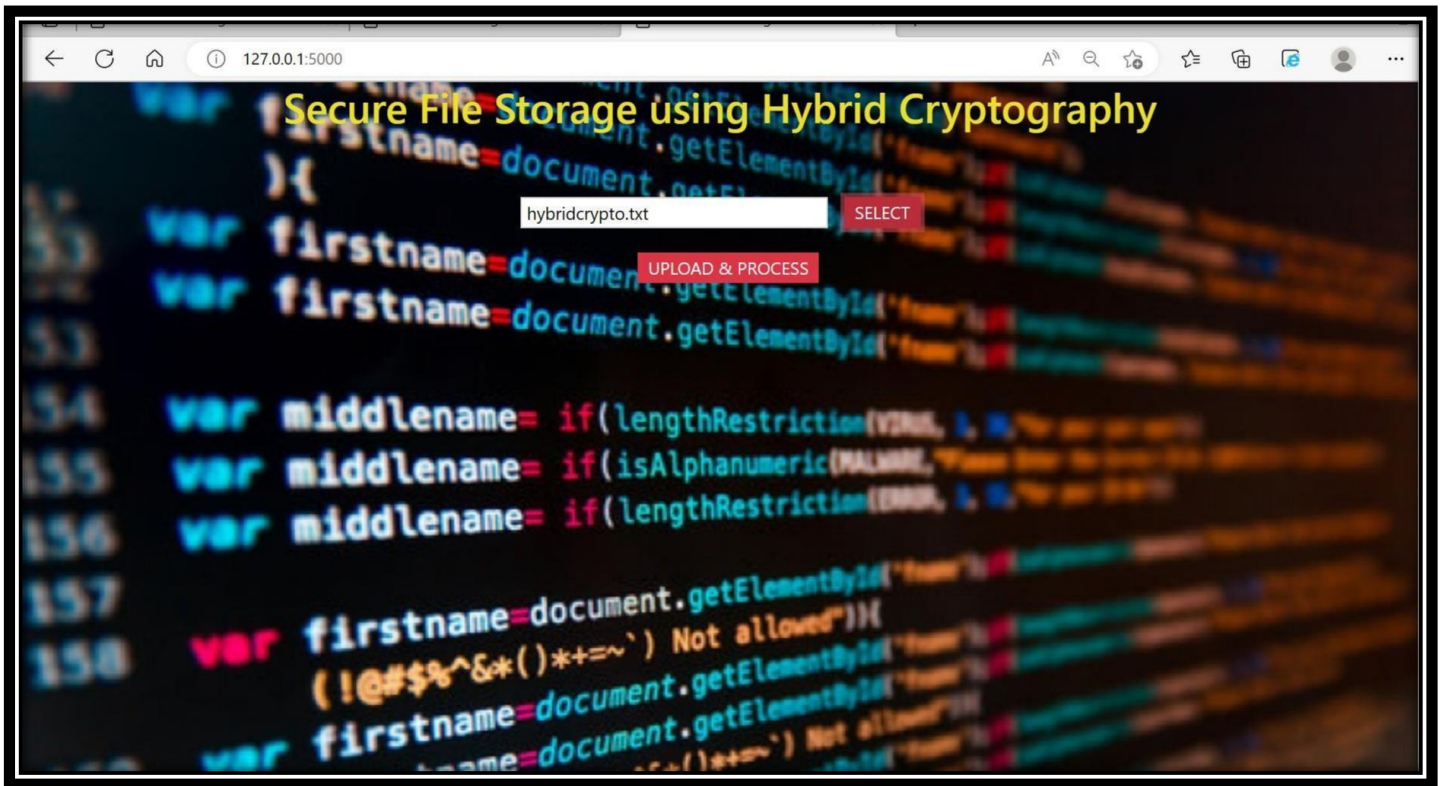
```
C:\Windows\System32\cmd.exe - python app.py
Microsoft Windows [Version 10.0.19044.2728]
(c) Microsoft Corporation. All rights reserved.

E:\wcn_exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6>python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 196-410-604
127.0.0.1 - - [26/Mar/2023 21:10:32] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Mar/2023 21:10:32] "GET /static/img/background.jpg HTTP/1.1" 200 -
127.0.0.1 - - [26/Mar/2023 21:10:34] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Mar/2023 21:10:34] "GET /static/img/background.jpg HTTP/1.1" 304 -
```

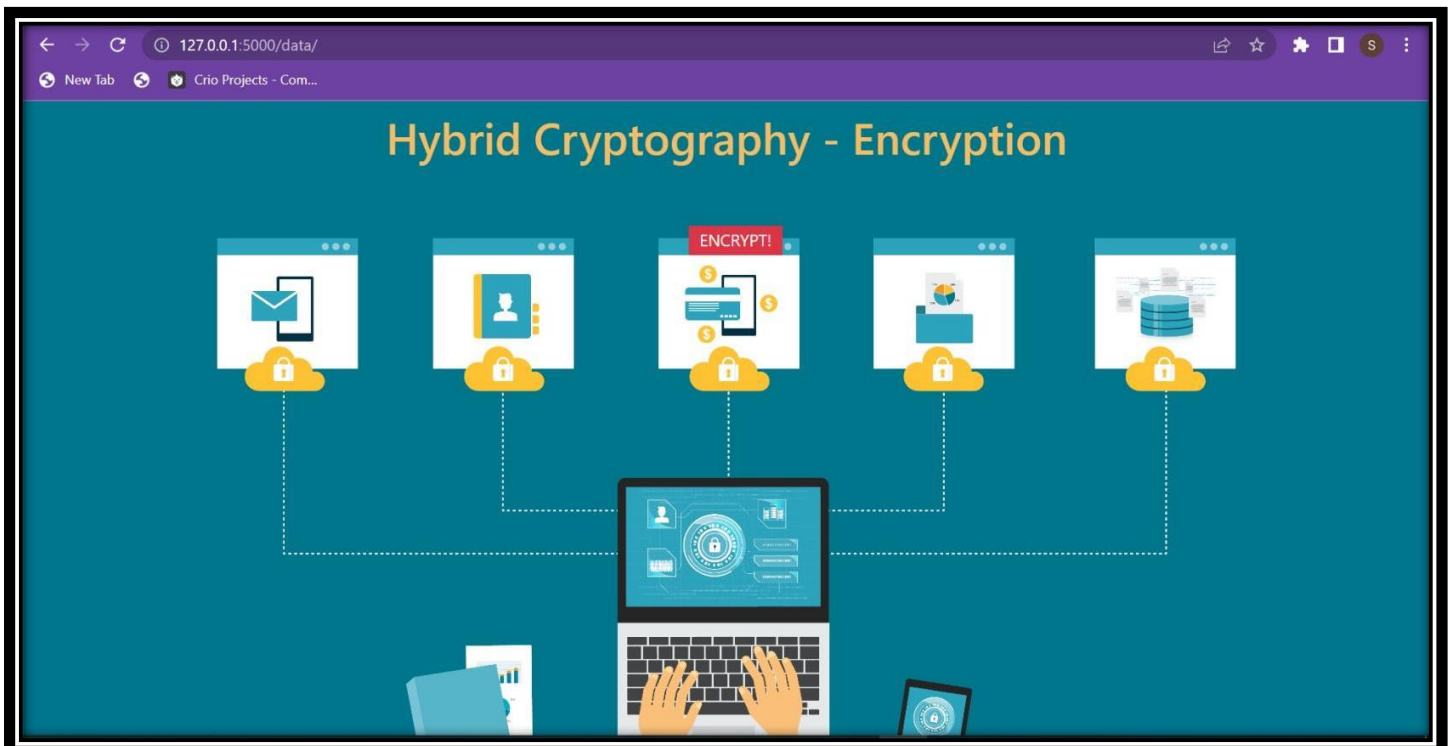
ORIGINAL FILE TO ENCRYPT & DECRYPT:-



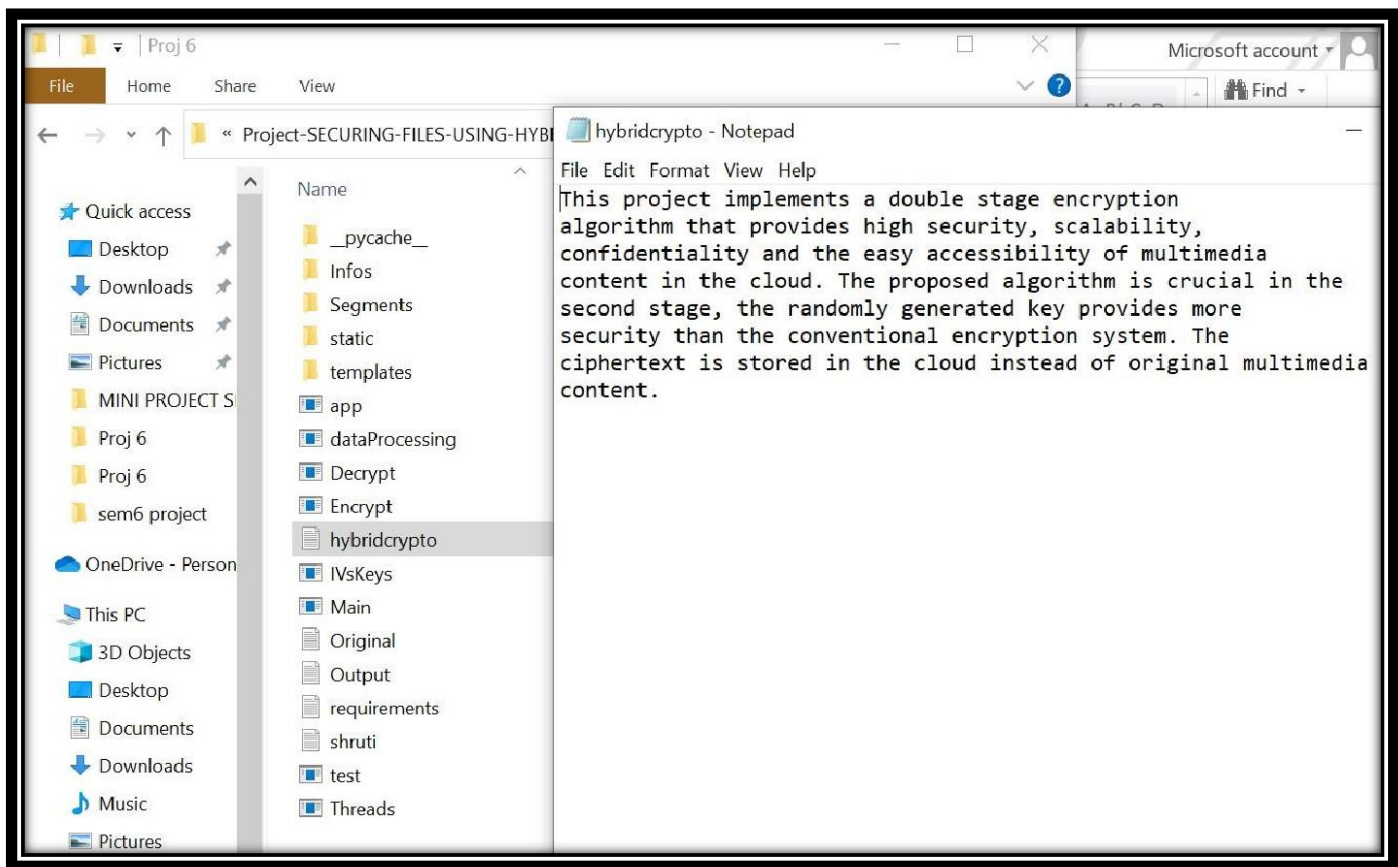
FILE UPLOADING & SELECTING PAGE:-



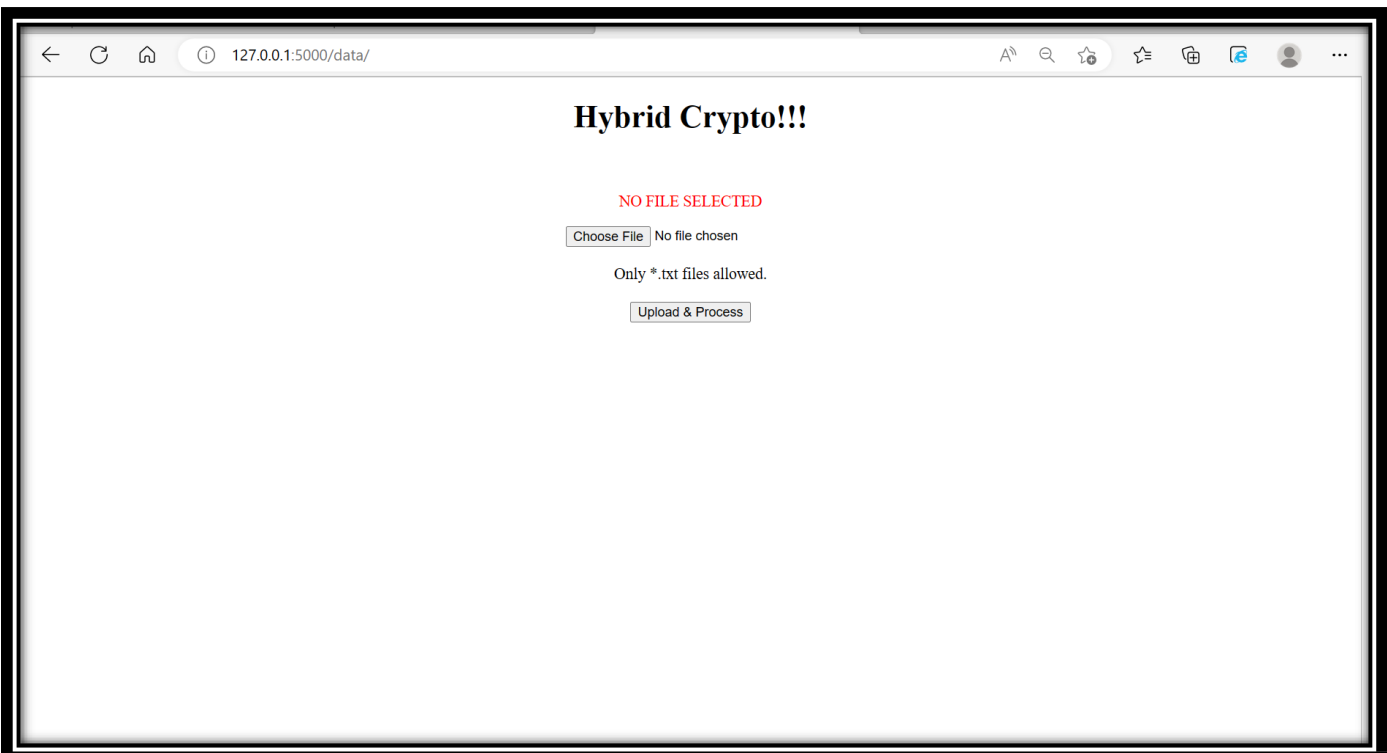
FILE ENCRYPTION PAGE:-



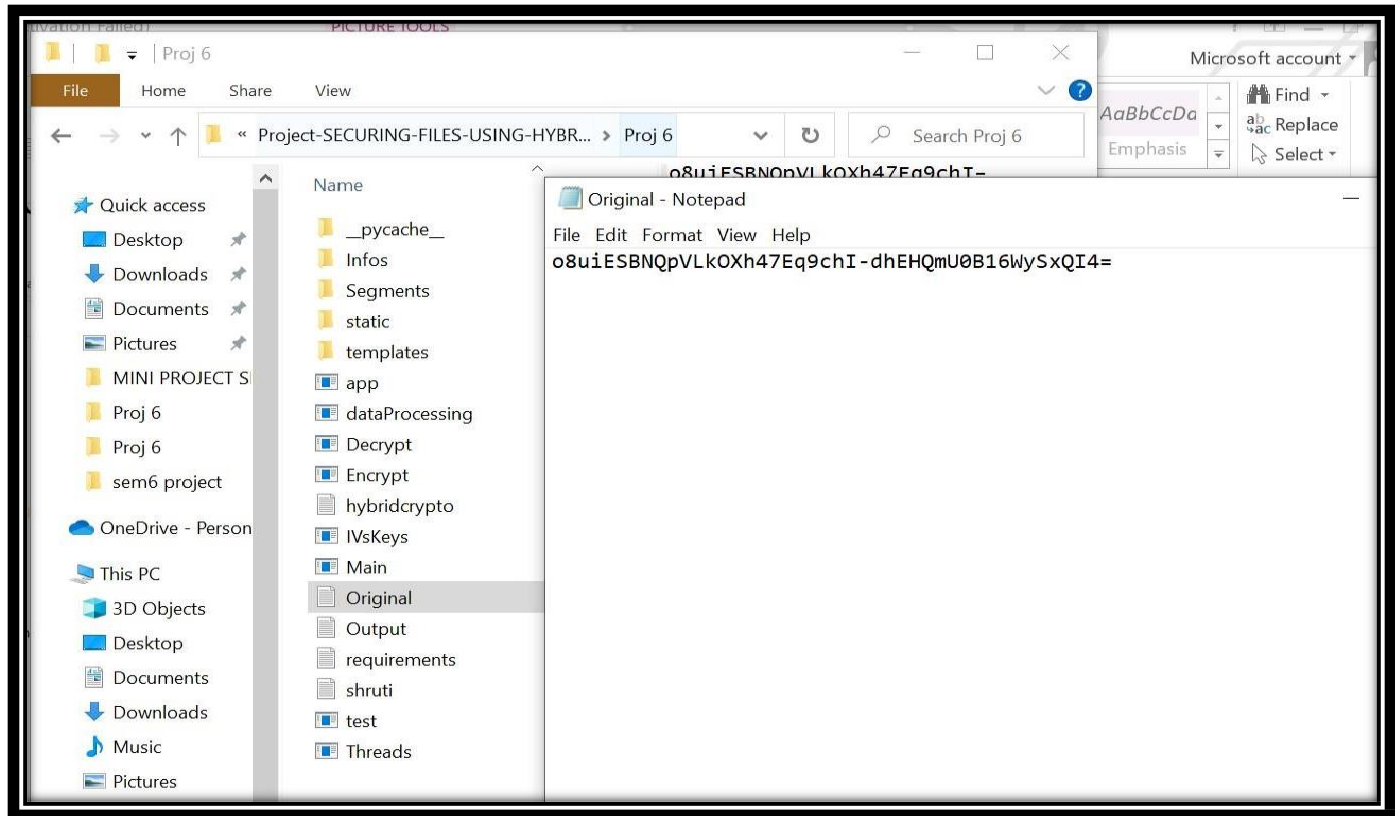
FILE TO ENCRYPT:-



IF NO FILE CHOSEN:-



ONCE FILE GETS ENCRYPTED:-



AFTER FILE ENCRYPTION:-

```
C:\Windows\System32\cmd.exe - python app.py
Microsoft Windows [Version 10.0.19044.2728]
(c) Microsoft Corporation. All rights reserved.

E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6>python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 196-410-604
127.0.0.1 - - [26/Mar/2023 21:28:29] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Mar/2023 21:28:29] "GET /static/img/background.jpg HTTP/1.1" 200 -
127.0.0.1 - - [26/Mar/2023 21:28:30] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [26/Mar/2023 21:28:48] "POST /data/ HTTP/1.1" 200 -
127.0.0.1 - - [26/Mar/2023 21:28:49] "GET /static/img/encryption.png HTTP/1.1" 200 -
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Encrypt.py:35: CryptographyDeprecationWarning: Blowfish has been deprecated
  cipher = Cipher(algorithms.Blowfish(key), modes.CBC(iv), backend=backend)
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Encrypt.py:74: CryptographyDeprecationWarning: IDEA has been deprecated
  cipher = Cipher(algorithms.IDEA(key), modes.CBC(iv), backend=backend)
['0.txt', '1.txt', '2.txt', '3.txt', '4.txt']
127.0.0.1 - - [26/Mar/2023 21:28:50] "GET /encrypt/ HTTP/1.1" 200 -
127.0.0.1 - - [26/Mar/2023 21:28:50] "GET /static/img/successful_encrypt.png HTTP/1.1" 200 -
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Infos\IV.txt
b'gAAAAABkIGu6xzYEZ0-Ot7VIGkInpjuYKmRN1ZMw6tUdr05mdYuaK3BwW8mDkJwwA3rxz8f0eHkTFFA-bWdIdWT6WWDW3qf_3J5au231rRnRDP6ncDiAbc='
IV.txt
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Infos\KEY1.txt
b'gAAAAABkIGu6n28PFduuZ_gmM1cE079Qo3Bc8fmqHwQyFwh6MdPvKeJXDUIVXD8SsqdPDV-16gCtds-wC-ORmIE057vCcSLU7mpMqI1zIG5EAR_NShyHitedtkwDoX1QM1mq00-519'
KEY1.txt
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Infos\KEY2.txt
b'gAAAAABkIGu6vYjIj_ATIVc0WxIyDhkPIG1STbxh8GcS4xNfMUCvW-tPHLkEq3JbAppi2qMLPkjtkiJ1DeGzZPsAwYKEOE---jk6JpNaUvYy9ywo2oHeOFA='
KEY2.txt
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Infos\Log.txt
b'gAAAAABkIGu6vYjIj_ATIVc0WxIyDhkPIG1STbxh8GcS4xNfMUCvW-tPHLkEq3JbAppi2qMLPkjtkiJ1DeGzZPsAwYKEOE---jk6JpNaUvYy9ywo2oHeOFA='
Log.txt
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Decrypt.py:24: CryptographyDeprecationWarning: Blowfish has been deprecated
  cipher = Cipher(algorithms.Blowfish(key), modes.CBC(iv), backend=backend)
```

```

C:\Windows\System32\cmd.exe - python app.py
127.0.0.1 - - [26/Mar/2023 21:28:48] "POST /data/ HTTP/1.1" 200 -
127.0.0.1 - - [26/Mar/2023 21:28:49] "GET /static/img/encryption.png HTTP/1.1" 200 -
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Encrypt.py:35: CryptographyDeprecationWarn
ing: Blowfish has been deprecated
cipher = Cipher(algorithms.Blowfish(key), modes.CBC(iv), backend=backend)
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Encrypt.py:74: CryptographyDeprecationWarn
ing: IDEA has been deprecated
cipher = Cipher(algorithms.IDEA(key), modes.CBC(iv), backend=backend)
['0.txt', '1.txt', '2.txt', '3.txt', '4.txt']
127.0.0.1 - - [26/Mar/2023 21:28:50] "GET /encrypt/ HTTP/1.1" 200 -
127.0.0.1 - - [26/Mar/2023 21:28:50] "GET /static/img/successful_encrypt.png HTTP/1.1" 200 -
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Infos\IV.txt
b'gAAAAABkIGu6xzYzE0-Ot7VIGknpjuNyKmrN1ZW6tUdr0SmdYuAK3BW8mDkJwwA3rxz8f8eHktFFA-bWdIdWt6WdW3qF_3J5au23IrNRDp6ncDiAbc='
IV.txt
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Infos\KEY1.txt
b'gAAAAABkIGu6tn8YS4LKMO1U9sbZCgFpk6Ywif52E_qLP108eInZ5IRVEDoDgoiG9xTC2FwdB8z4IDbXgow204ZVMo1cYD-9hvhHs9DBnUeot_0Rbkl-fcz30='
KEY1.txt
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Infos\KEY2.txt
b'gAAAAABkIGu6n28PfduuZ_gmM1cE079Qo3Bc8fmqHwQvyFwh6MdPvKeJXDUIVxD8SsqdPDV-16gCtds-wC-ORmIE057vCCsLU7mpMqIlzNIG5EAR_NShyHitedtkwDoXlQM1mQoO-519'
KEY2.txt
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Infos\Log.txt
b'gAAAAABkIGu6vyJij_AT1Vc0WxiYDhkP1G1STbxh8GcS4xNfMUCvW-tPHLkEq3JbAppi2qMLPkjtkiJ1DeGzZPsAwYKEOE---jk6JpNAuVvy9ywo2oHeOFYA='
Log.txt
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Decrypt.py:24: CryptographyDeprecationWarn
ing: Blowfish has been deprecated
cipher = Cipher(algorithms.Blowfish(key), modes.CBC(iv), backend=backend)
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Decrypt.py:48: CryptographyDeprecationWarn
ing: IDEA has been deprecated
cipher = Cipher(algorithms.IDEA(key), modes.CBC(iv), backend=backend)
0.03870892524719238
From Encrypted file - 0 -> This project implements a double stage encryption
algorithm that provides high s
From Encrypted file - 1 -> ecurity, scalability,
confidentiality and the easy accessibility of multimedia
c
From Encrypted file - 2 -> ontent in the cloud. The proposed algorithm is crucial in the
second stage, the
From Encrypted file - 3 -> randomly generated key provides more
security than the conventional encryption s
From Encrypted file - 4 -> ystem. The
ciphertext is stored in the cloud instead of original multimedia

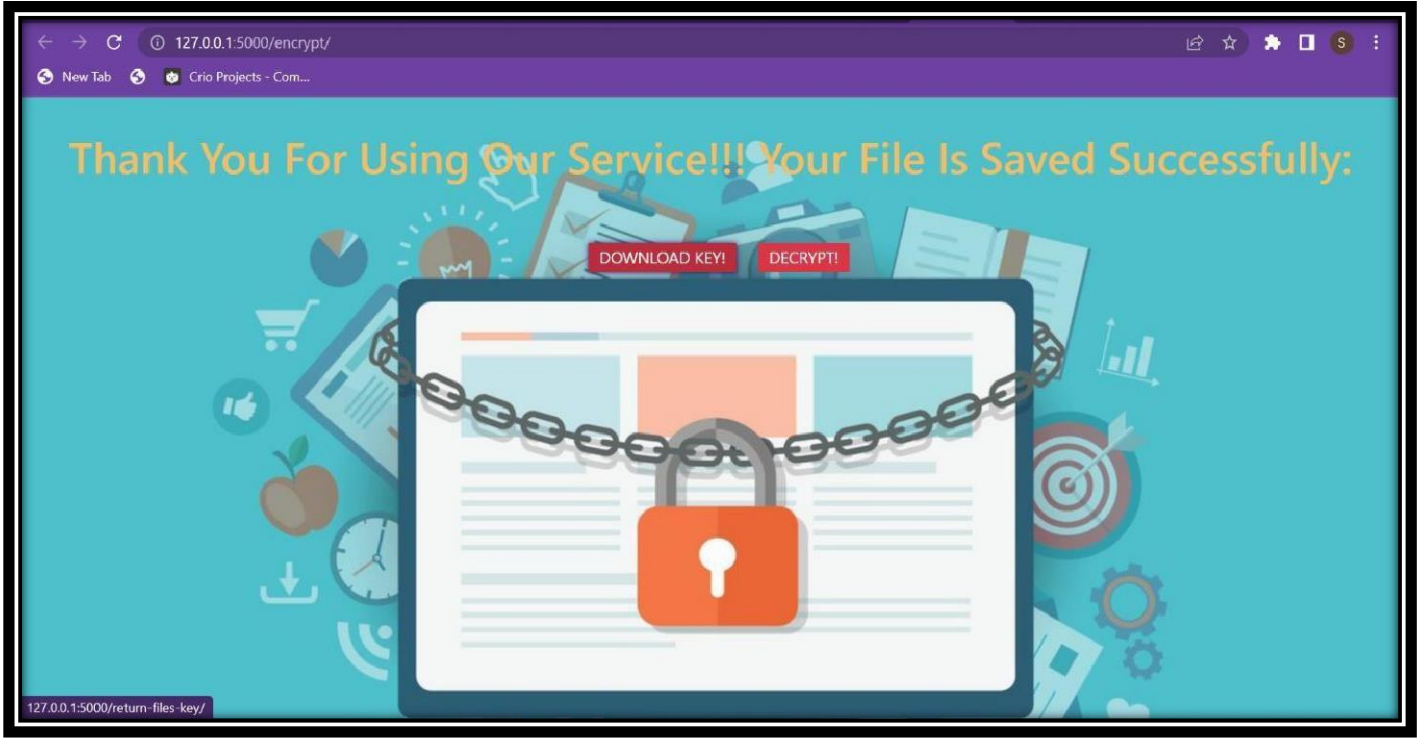
```

```

C:\Windows\System32\cmd.exe - python app.py
b'gAAAAABkIGu6n28PfduuZ_gmM1cE079Qo3Bc8fmqHwQvyFwh6MdPvKeJXDUIVxD8SsqdPDV-16gCtds-wC-ORmIE057vCCsLU7mpMqIlzNIG5EAR_NShyHitedtkwDoXlQM1mQoO-519'
KEY2.txt
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Infos\Log.txt
b'gAAAAABkIGu6vyJij_AT1Vc0WxiYDhkP1G1STbxh8GcS4xNfMUCvW-tPHLkEq3JbAppi2qMLPkjtkiJ1DeGzZPsAwYKEOE---jk6JpNAuVvy9ywo2oHeOFYA='
Log.txt
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Decrypt.py:24: CryptographyDeprecationWarn
ing: Blowfish has been deprecated
cipher = Cipher(algorithms.Blowfish(key), modes.CBC(iv), backend=backend)
E:\wcn exp\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Project-SECURING-FILES-USING-HYBRID-CRYPTOGRAPHY-master\Proj 6\Decrypt.py:48: CryptographyDeprecationWarn
ing: IDEA has been deprecated
cipher = Cipher(algorithms.IDEA(key), modes.CBC(iv), backend=backend)
0.03870892524719238
From Encrypted file - 0 -> This project implements a double stage encryption
algorithm that provides high s
From Encrypted file - 1 -> ecurity, scalability,
confidentiality and the easy accessibility of multimedia
c
From Encrypted file - 2 -> ontent in the cloud. The proposed algorithm is crucial in the
second stage, the
From Encrypted file - 3 -> randomly generated key provides more
security than the conventional encryption s
From Encrypted file - 4 -> ystem. The
ciphertext is stored in the cloud instead of original multimedia
content.
0.01449131965637207
127.0.0.1 - - [26/Mar/2023 21:29:07] "GET /decrypt/ HTTP/1.1" 200 -
127.0.0.1 - - [26/Mar/2023 21:29:07] "GET /static/img/successful_decrypt.png HTTP/1.1" 304 -
127.0.0.1 - - [26/Mar/2023 21:29:08] "GET /return-files-data/ HTTP/1.1" 200 -

```

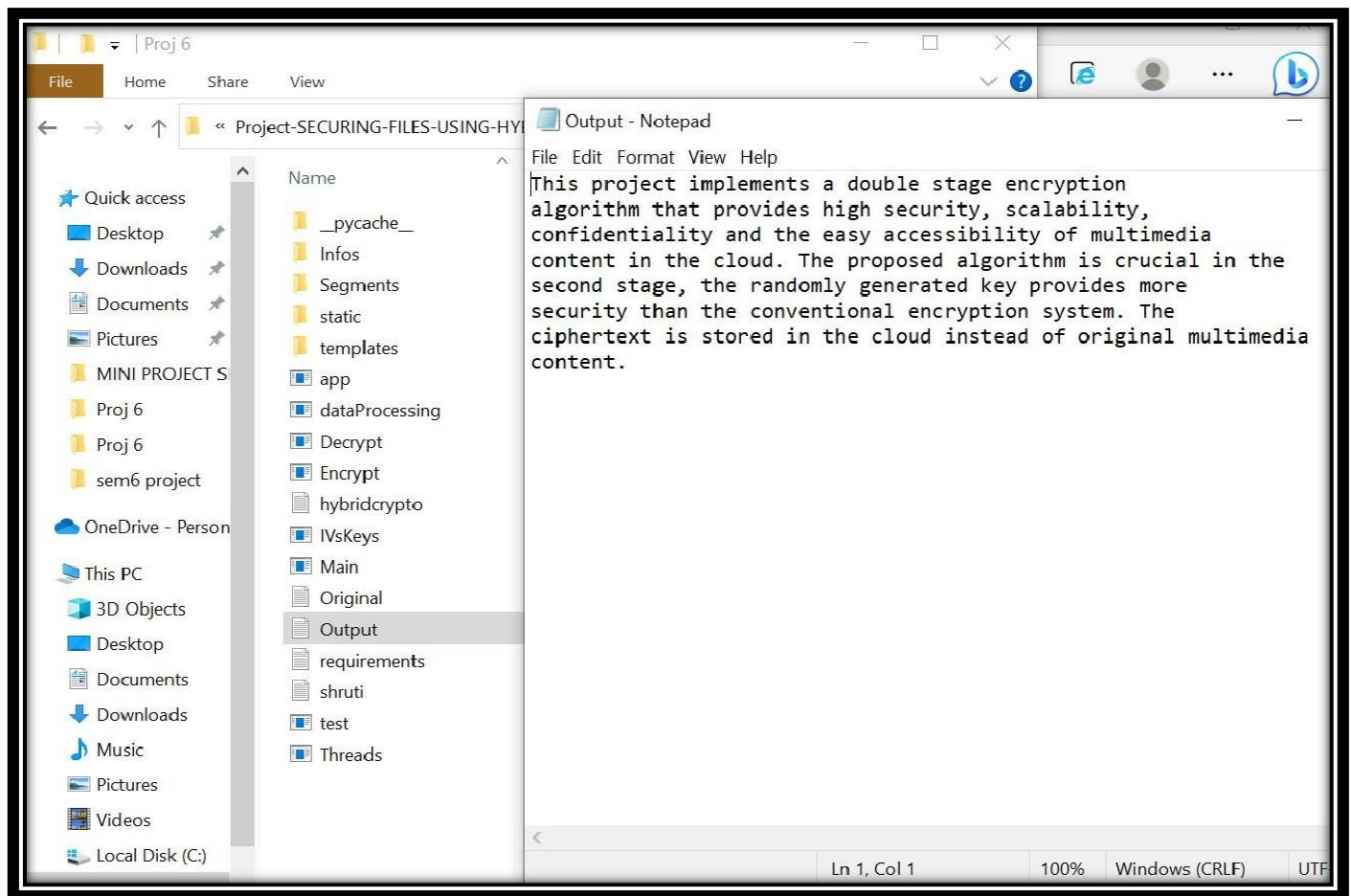
TO DECRYPT FILE:-



ONCE FILE GETS DECRYPTED:-



DECRYPTED FILE:- We got original final i.e. (deciphered file)



CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION:-

- ▶ The main aim of this system is to securely store and retrieve data that is only controlled by the owner of the data.
- ▶ Storage issues of data security are solved using cryptography and steganography techniques.
- ▶ In the future we can add public key cryptography to avoid any attacks during the transmission of the data from the client to the server.
- ▶ This project implements a double stage encryption algorithm that provides high security, scalability, confidentiality and the easy accessibility of multimedia content.
- ▶ The proposed algorithm is crucial in the second stage, the randomly generated key provides more security than the conventional encryption system.
- ▶ The cipher text is stored in the cloud instead of original multimedia content. The cipher text is undoubtedly hard to recover the original content for random asymmetric key.
- ▶ Wide application of the proposed algorithm protects the information from the side channel attacker to grab the multimedia data.

This project implements a double stage encryption algorithm that provides high security, scalability, confidentiality and the easy accessibility of multimedia content in the cloud.

The proposed algorithm is crucial in the second stage, the randomly generated key provides more security than the conventional encryption system. The cipher text is stored in the cloud instead of original multimedia content. The cipher text is undoubtedly hard to recover the original content for random asymmetric key. Wide application of the proposed algorithm protects the information from the side channel attacker to grab the multimedia data from the cloud. Thus, the multimedia content is safe in the cloud.

7.2 Future Scope:-

In the world of data being the key asset, safeguarding our asset is primary responsibility. Communications, databases, infrastructure, transactions, knowledge; an organization's data is arguably its most valuable asset. It is in a business' best interests to keep its information safe, regardless of legal or regulatory requirements.

- A system which stores data after encryption.
- This prevents data leak if a breach occurred.
- Any form of data can be stored.
- It ensures data confidentiality to users.

This proposed hybrid algorithm can be made much more powerful and secure by increasing the number of iterations in the encryption algorithm to suit the level of security required. An inverse policy of reducing the number of iterations for lower security can also be employed. We can also go for combining another algorithm that will encrypt data given by the IDEA algorithm.

This inclusion of third algorithm will increase the security but there are two phase of a coin. As a result Security will increase but time that is taken to convert the plain text into final cipher text will be greater than previous hybrid algorithm. So it is the demand of application in which you are going to use security algorithm which factor is important time or security. We must play a fair role between time taken by the algorithm and level of security, both must be reasonable.

Acknowledgement

As every project is ever complete with the guidance of experts. So we would like to take this opportunity to thank all those individuals who have contributed in visualizing this project.

We express our deepest gratitude to our project guide Prof Swati Vyas (CSE (AIML))Department, Smt. IndiraGandhi College of Engineering, University of Mumbai) for her valuable guidance, moral support and devotionbestowed on us throughout our work.

We would also take this opportunity to thank our project coordinator **Prof. SWATI VYAS** for her guidance in selecting this project and also for providing us all the details on proper presentation of this project.

We extend our sincere appreciation to our entire professors from Smt. Indira Gandhi College of Engineering for their valuable inside and tip during the designing the project. Their contributions have been valuable in many ways that we find it difficult to acknowledge them individually.

We are also grateful to our HOD **Prof. SONALI DESHPANDE** for extending his help directly and indirectly through various channels in our project.

If I can say in words I must at the outset my intimacy for receipt of affectionate care to Smt. Indira GandhiCollege of Engineering for providing such a simulating atmosphere and wonderful work environment.

References:-

- i. Sombir Singh, Sunil k. Maakar, Dr. Sudesh Kumar “Enhancing the Security of DES Algorithm Using Transposition Cryptography Techniques”,IJARCSSE, Volume 3, Issue 6, pp 464-470, June 2013.
- ii. Nick Hoffman “A Simplified IDEA Algorithm” Department of Mathematics,Northern Kentucky University pp 1-5, 2007.
- iii. Meier, W., On the Security of the IDEA block cipher, Advances in Cryptology.
- iv. Atul Kahate “Cryptography and Network Security” second edition
- v. Shaaban Sahmoud, Wisam Elmasry and Shadi Abdulfa “Enhancement the security of AES againt modern attacks by using variable key block cipher”
- vi. Data Encryption Standard (DES), Federal Information processing standards,
- vii. Publication 46-3, 1999 October 25
- viii. Advanced Encryption Standard, National Institute of Standards and Technology (US), [URL:http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf](http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf)
- ix. William Stallings:”Cryptography and network security:Principles and Practices”.
- x. Advanced Encryption Standard, [online], Available:
- xi. [URL:http://en.wikip/Advanced_Encryption_Standar](http://en.wikip/Advanced_Encryption_Standar)
[d](http://en.wikip/Advanced_Encryption_Standar)