A Mini Project Report on

Secure File Storage Using Hybrid Cryptography

Submitted in partial fulfillment of the requirements for the award of
the degree of

Bachelor of Engineering

in

Computer Engineering

by

Siddharth Vijay Bhamare (06)
Pushpak Shaligram Maind (28)
Rahul Sudhir Patil (37)
Vishal Pradeep Shirke (48)

Under the Guidance of

Pro. Ranjana Singh

Department of Computer Engineering
Watumull Institute of Electronics Engineering and Computer Technology, Ulhasnagar
UNIVERSITY OF MUMBAI

Academic Year 2021-2021

Approval Sheet

This Mini Project Report entitled ***Secure File Storage Using Hybrid Cryptography*** Submitted by ***Siddharth Bhamare (06), Pushpak Maind (28), Rahul Patil (37), Vishal Shirke (48)*** is approved for the partial fulfillment of the requirement for the award of the degree of ***Bachelor of Engineering*** in ***Computer Engineering*** from ***University of Mumbai***.

(Prof. Ranjana Singh)

Guide

Head Department of Computer Engineering

Place:

Date:

CERTIFICATE

This is to certify that the mini project entitled ***"Secure File Storage Using Hybrid Cryptography"*** submitted by ***Siddharth Bhamare (06), Pushpak Maind (28), Rahul Patil (37), Vishal Shirke (48)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Computer Engineering***,to the University of Mumbai, is a bonafide work carried out during academic year 2021-2022.

(Prof. Ranjana Singh)

Guide Name & Signature

Examiners:

1.

2.

Head Department of Computer Engineering

Principal

Place:

Date:

Declaration

We declare that this written submission represents our 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 idea/data/fact/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.

_____

(Signature)

_____

Siddharth Bhamare(06)
Pushpak Maind(28)
Rahul Patil(37)
Vishal Shirke(48)

Date:

Abstract

The 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 also have some drawbacks. To improve the strength of these algorithms, we propose a new hybrid cryptographic algorithm in this paper. The algorithm is designed using a combination of three symmetric and one general purpose cryptographic techniques. These three primitives can be achieved with the help of Data Encryption Standard (DES), Advanced Encryption Standard (AES), International Data Encryption Standard (IDEA) and the one Blowfish algorithm. This new hybrid cryptographic algorithm has been designed for better security with integrity.

Keywords: Cryptography, Encryption, Decryption, DES, AES, IDEA, Blowfish algorithm.

Contents

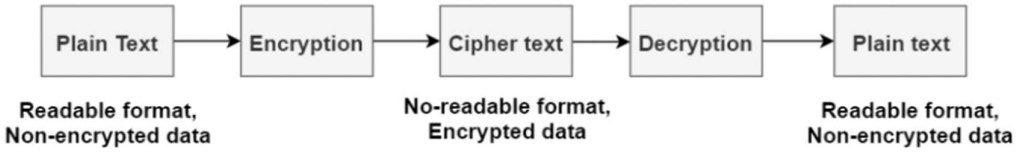
1.Introduction

1.1. Problem Definition:

Traditional storage devices such as flash drives, hard disks and other kinds of physical storage devices are slowly becoming obsolete. The reason for this is that, on the business front, global expansion of companies requires data to be shared amongst employees for collaborative working. On the user's personal usage front, many users nowadays have multiple devices, such as one or more mobile/cell phones, tabs, laptops, desktop PCs etc. Hence cloud storage provides a way to access one's personal data across all of one's personal devices. Hence more and more people are shifting towards the more convenient option of cloud for storing their data. The ability to access files from remote locations using just a stable internet connection gives cloud an edge over other storage options, but on cloud, multiple users can access your data (shared data), so data integrity and data security are not maintained as we expect. In this case, the cryptography concept was introduced.

1.2. Cryptography:

Cryptography is the method of transmitting secured data and communications via a few codes so that only the destined person knows about the actual information that is transmitted.



Today, we all are going through a digital revolution and this digital revolution has brought many changes in technologies. Such as growth in usage of digital computation is increasing exponentially. Also, companies are growing rapidly and opening offices at various locations across the globe. As they are expanding globally they also need to access users data from any location for which they are asking permissions from users. Due to which some problems are occurring like the storage has security risks, data leaking is possible, etc. and for us the security of our data is more important isn't it?

1.3. Problem Domain:

As we know there are various issues that DES is no longer 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. Furthermore, AES is an algorithm famous for its speed of execution and security. Also we consider IDEA, that in the future the speed of computers will enhance, 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, AES, IDEA along with the Blowfish algorithm. So this hybrid system would have combined security of all four algorithms.

1.4.Problem Solution:

A Computer Network is an interconnected group of autonomous computing nodes, which use a well defined, mutually agreed set of rules and conventions known as protocols, to interact with one-another meaningfully and allow resource sharing preferably in a predictable and controllable manner. Communication has a major impact on today‟s business. It is desired to communicate data with high security. With the rapid development of network technology, internet attacks are also versatile, the traditional encryption algorithms (single data encryption) is not enough for today‟s information security over the internet, so we propose this hybrid Cryptography Algorithm. It is a design for transferring data with better security. At present, various types of cryptographic algorithms provide high security to information on networks, but there are also some drawbacks. This hybrid algorithm is designed for better security by combinations of DES, AES, IDEA and Blowfish.

1.5.Objective:

This hybrid algorithm has high security of data transmission over the network. This whole work is focused on how we can increase the security of data transmission. Security is necessary when we transmit highly sensitive data such as Banking transactions, Military information and many more.This hybrid algorithm fulfills these criteria up to the mark. This work results in more secure transmission of data comparatively to DES, IDEA and AES data encryption algorithms.

1.6. Scope:

Nowadays, data has become the key asset for everyone, and keeping our asset safe is our primary responsibility. Communications, databases, infrastructure, transactions, knowledge contains organization’s data, which is its most valuable asset. We all want a system that keeps information safe and which follows regardless of legal or regulatory requirements. Such as a system which stores data after encrypting it. Because of which it is possible to prevent data leak if breach occurred. And a system that stores data in any form. And if a system contains all these things then it will ensure data confidentiality to the users.

2. Existing System/Project

Hybrid Cryptography concept is used for securing storage systems of the 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.
2.1 To make the centralized cloud storage secure ECC (Elliptic Curve Cryptography) algorithm is implemented. This approach uses a single key for encryption and decryption and the complete process takes place at the client side. This methodology performs steps such as: a. Authentication, b. Key generation operation, c. Encryption, d. Decryption.

2.2 In this proposed system three step procedure is used. Firstly, Diffie Hellman is used for exchanging keys. Thereafter authentication is performed using a digital signature scheme. Finally data is encrypted using AES and then uploaded to the required cloud system. For decryption the reverse procedure is implemented.

2.3 Combination of RSA algorithm and MD5 to assure various security measures such as confidentiality, data integrity, nonrepudiation etc. It uses RSA key generation algorithm for generation of encrypted key for encryption and decryption
2.4 process. MD5 digest is used for accepting an input of length up to 128 bit and processing it and generating an output of padded length for encryption and decryption process.

2.5 Implementation of Trusted Storage System using Encrypted File System (EFS) and NTFS file system drive with help of cache manager for securing data files. EFS encrypts stored files by automatically using cryptographic systems. The process takes place as follows, firstly the application writes files to NTFS which in turn places in cache and return backs to NTFS. After this NTFS asks EFS to encrypt files and heads them towards the disk.

2.6 Cloud Storage Security Service is provided by using separate servers viz. User Input, Data Storage and User Output. Three different servers are used to ensure that failure of any of the servers doesn't harm the data. User Input server is used for storing user files and input data by providing user authentication and making sure the data is not accessed by any of the unauthorized means. Data storage server is the place where the encryption using AES is performed to secure user input and the encrypted files are transferred to the User Output server. User Output Server is the place from where the user gets the output file or the decrypted file and use it for further use.

3. Technology Stack

Hardware Requirements:

- Computer/laptop
- Intel i3 6th Gen or higher/4gb Ram

Software Requirements:

- Python
- Flask
- HTML
- CSS
- JavaScript

Python Libraries:

- cryptography
- Flask
- Werkzeug
- gunicorn

4.Benefits and Application

4.1 Benefits :

●**Secrecy:** Nobody will be able to get any information about the encrypted plaintext (except the length), unless they have access to the secret key.
●**Randomization:** The encryption is randomized. Two messages with the same plaintext will not yield the same ciphertext. This prevents attackers from knowing which ciphertext corresponds to a given plaintext.
●**Harder to Crack:** Using 5 different algorithms leaves complications for attackers to guess the correct one, Also method of implementation can be different.
●**Significant increase in amount of keys:** There are 4 keys used plus 1 key to encrypt every key which further complicates the attacker.

4.2 Application:

Today, there are hundreds of security solutions available in many market areas, ranging from Financial Services, and Broadcasting to Government. AES is the name of a proven, secure, and universally applicable block encryption algorithm, which permits effective protection of transmitted and stored data against unauthorized access by third parties. The fundamental criteria for the development of Hybrid algorithm were highest security requirements along with easy hardware and software implementation for fast execution. This Hybrid algorithm can easily be embedded in any encryption software. Data encryption can be used to protect data transmission and storage. Typical fields are: Audio and video data for cable TV, pay TV, video conferencing, distance learning, business TV, VoIP. There are various fields in which this Hybrid algorithm can be used. These are as follows:-

A. Sensitive financial and commercial data.
B. Email via public networks.
C. Transmission links via modem, router or ATM link, GSM technology.
D. Smart cards.
E. Antivirus in computers and laptops.
F. Banking transactions.
G. Military information.

Encryption results in easy detection and recovery of the key. However, since there are 2192 possible keys, this result has no impact on the practical security of the cipher for encryption provided the encryption keys are chosen at random. AES is generally considered to be a very secure cipher and both the cipher development and its theoretical basis have been openly and widely discussed so this Hybrid algorithm will result in higher security. AES is a patented and universally applicable block encryption algorithm, which permits the effective protection of transmitted and stored data against unauthorized access by third parties. With a key of 128 bits in length, AES is far more secure than the widely known DES based on a 56-bit key. The fundamental criteria for the development of AES were military strength for all security requirements and easy hardware and software implementation. The algorithm is used worldwide in various banking and industry applications. They predestine the algorithm for use in a great number of commercial application

5. Project Design

5.1. Proposed System

- A secure file and password storage using cryptography.
- A proper registration/login-based system for using the website.
- The system will help user to secure data using different encryption algorithm.
- User can upload passwords as well as text files.
- The user with the proper key will be able to access the particular file.

The primary goal of the system is to provide and simulate a solution to face the challenges and solve security issues that exists in cloud computing.

5.2. DES

DES is the archetypal block cipher — an algorithm that takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another ciphertext bit string of the same length. DES also uses a key to customize the transformation, so that decryption can only be performed by those who know the particular key used to encrypt.

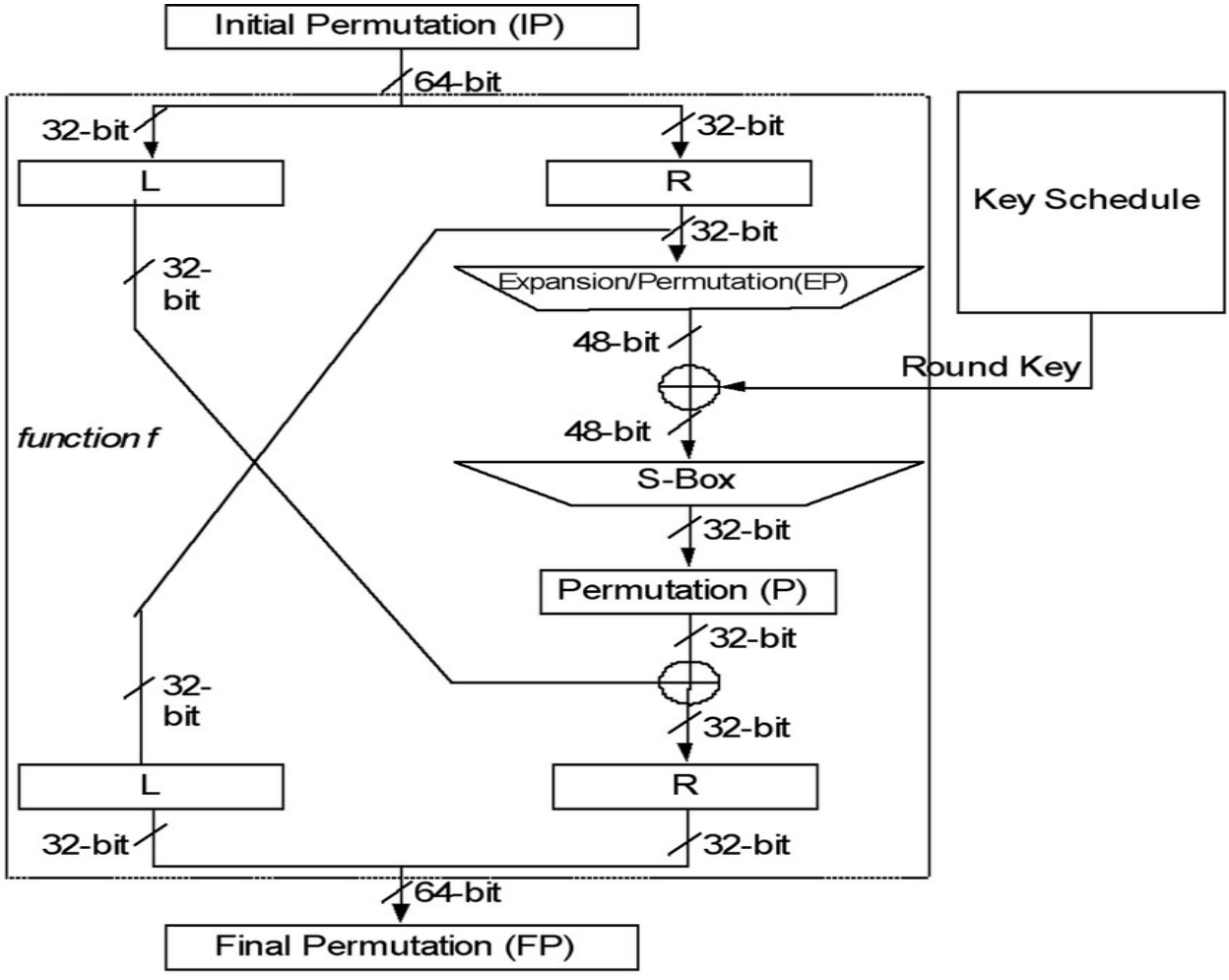
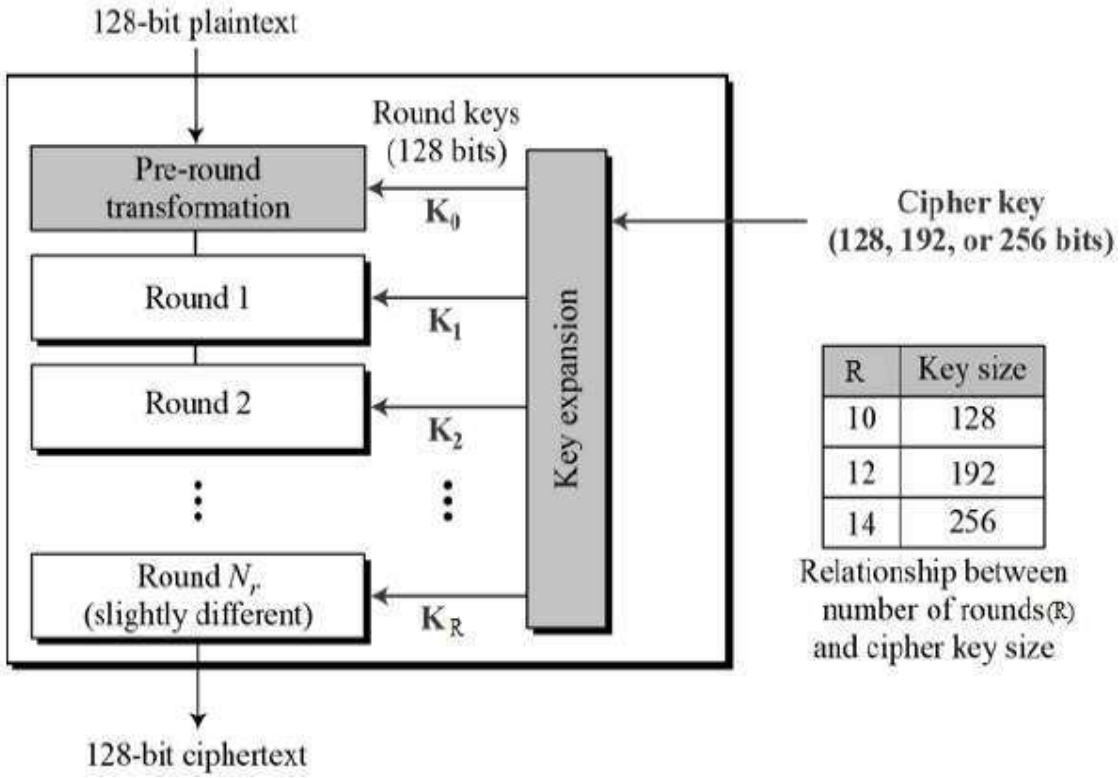


Fig. Block Diagram of Data Encryption Standard

In the case of DES, the block size is 64 bit, but only 56 bits are used and the remaining 8 bits can be used for parity, and then discarded in the algorithm. Therefore, the effective key length of DES is 56 bit. The algorithm‟s overall structure is shown in Fig. there are 16 identical same processes; termed rounds. There is also an initial and final permutation, known as IP and FP (the FP is inverse function of IP (IP ―revocation‖ FP operations, and vice versa)). Before the main rounds, the block is divided into two 32 bit half blocks and processed at same times; this crossing process is known as the Feistel scheme. Feistel scheme is used to ensure the similarity of both the encryption and decryption processes. The only difference is the sub-key, which is reversed and used in the decryption process and the remaining part is the same. This design simplifies the algorithm implementation, especially for hard implementation. The symbol denotes the (XOR) operation. The ―F- function‖ scrambles data process with one sub-key, then the output from F-function doing the XOR operation with other half block data, and the halves are swapped before the next round. After the final round, the halves are not swapped; this is a feature of the Feistel structure which makes encryption and decryption similar processes.

5.3. AES

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 a 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.



R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

Fig. Block Diagram of Advanced Encryption Standard

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 round key 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:
(a) Sub Bytes - according to a lookup table each byte is replaced with another in a non-linear substitution step.
(b) Shift Rows - a transposition step where the last 3 rows of the state are shifted cyclically a certain number of steps.
(c) Mix Columns - a mixing operation which operates on the columns of the state, combining the 4 bytes in each column.
(d) Add Round Key.

Final Round (no Mix Columns):
(a) Sub Bytes.
(b) Shift Rows.
(c) Add Round Key.

5.4. IDEA

International Data Encryption Algorithm (IDEA) is a block cipher designed by Xuejia Lai and James L. Massey of ETH-Zürich and was first described in 1991. The block cipher IDEA operates with 64-bit plaintext and ciphertext blocks and is controlled by a 128-bit key. The fundamental innovation in the design of this algorithm is the use of operations from three different algebraic groups. The substitution boxes and the associated table lookups used in the block ciphers available to-date have been completely avoided. The algorithm structure has been chosen such that, with the exception that different key sub-blocks are used, the encryption process is identical to the decryption process.

Key Generation:
The 64-bit plaintext block is partitioned into four 16-bit sub-blocks, since all the algebraic operations used in the encryption process operate on 16-bit numbers. Another process produces for each of the encryption rounds, six 16-bit key sub-blocks from the 128-bit key. Since a further four 16-bit key-sub-blocks are required for the subsequent output transformation, a total of 52 (= 8 x 6 + 4) different 16-bit sub-blocks have to be generated from the 128-bit key.\
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 resulting 128-bit block is again partitioned into eight 16-bit sub-blocks to be directly used as the next eight key sub-blocks.
iii. The cyclic shift procedure described above is repeated until all of the required 52 16-bit key sub-blocks have been generated.

Encryption:

The functional representation of the encryption process is shown in Figure 1. The process consists of eight identical encryption steps (known as encryption rounds) followed by an output transformation. The structure of the first round is shown in detail. In the first encryption round, the first four 16-bit key sub-blocks are combined with two of the 16-bit plaintext blocks using addition modulo 2^{16}, and with the other two plaintext blocks using multiplication modulo $2^{16} + 1$. The results are then processed further as shown in Figure 1, whereby two more 16-bit key sub-blocks enter the calculation and the third algebraic group operator, the bit-by-bit exclusive OR, is used. At the end of the first encryption round four 16-bit values are produced which are used as input to the second encryption round in a partially changed order. The process described above for round one is repeated in each of the subsequent 7 encryption rounds using different 16-bit key sub-blocks for each combination. During the subsequent output transformation, the four 16-bit values produced at the end of the 8^{th} encryption round are combined with the last four of the 52 key subblocks using addition modulo 2^{16} and multiplication modulo $2^{16} + 1$ to form the resulting four 16-bit ciphertext blocks.

Decryption:

The computational process used for decryption of the ciphertext is essentially the same as that used for encryption of the plaintext. For plaintext exceeding decryption, different 16-bit key sub-blocks are generated.More precisely, each of the 52 16-bit key sub-blocks used for decryption is the inverse of the key sub-block used during encryption in respect of the applied algebraic group operation. Additionally, the key sub-blocks must be used in the reverse order during decryption in order to reverse the encryption process. IDEA supports all modes of operation as described by NIST in its publication FIPS 81. A block cipher encrypts this fixed size, the simplest approach is to partition the plaintext into blocks of equal length and encrypt each separately. This method is named Electronic Code Book (ECB) mode. However, Electronic Code Book is not a good system to use with small block sizes (for example, smaller than 40 bits) and identical encryption modes. As ECB has disadvantages in most applications, other methods named modes have been created. They are Cipher Block Chaining (CBC), Cipher Feedback (CFB) and Output Feedback (OFB) modes.

5.5. BLOWFISH

Blowfish is a symmetric block encryption algorithm designed which is fast, compact, simple and secure to use as: 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 a 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 round Feistel network and this algorithm is divided into two parts.
1. Key-expansion.
2. Data Encryption.

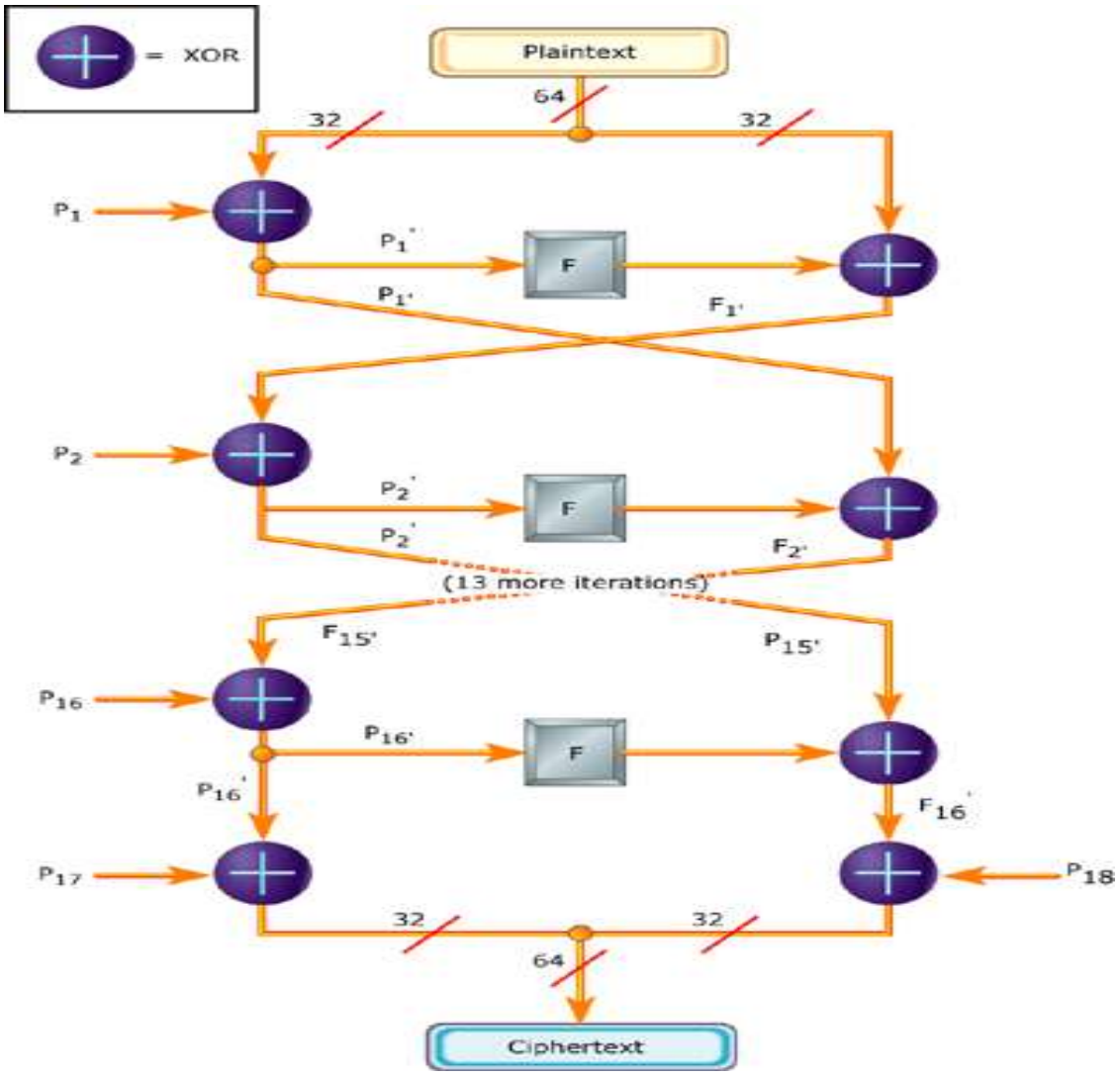


Fig. Block Diagram of BlowFish Encryption Algorithm

1. 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.

1. Data Encryption: It has a function to iterate 16 times of the 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.

5.6. FERNET

Cryptography is the practice of securing useful information while transmitting from one computer to another or storing data on a computer. Cryptography deals with the encryption of plaintext into ciphertext and decryption of ciphertext into plaintext. Python supports a cryptography package that helps us encrypt and decrypt data. The fernet module of the cryptography package has inbuilt functions for the generation of the key, encryption of plaintext into ciphertext, and decryption of ciphertext into plaintext using the encrypt and decrypt methods respectively. The fernet module guarantees that data encrypted using it
cannot be further manipulated or read without the key.

Methods Used:

generate_key() : This method generates a new fernet key. The key must be kept safe as it is the most important component to decrypt the ciphertext. If the key is lost then the user can no longer decrypt the message. Also if an intruder or hacker gets access to the key they can not only read the data but also forge the data.

encrypt(data) : It encrypts data passed as a parameter to the method. The outcome of this encryption is known as a "Fernet token" which is basically the ciphertext. The encrypted token also contains the current timestamp when it was generated in plaintext. The encrypt method throws an exception if the data is not in bytes.
Parameters:
data (bytes) – The plaintext to be encrypted.
Return value: A ciphertext that cannot be read or altered without the key. It is URL-safe base64-encoded and is referred to as a Fernet token.

decrypt(token,ttl=None) : This method decrypts the Fernet token passed as a parameter to the method. On successful decryption the original plaintext is obtained as a result, otherwise an exception is thrown.
Parameters:
1. token (bytes) – The Fernet token (ciphertext) is passed for decryption.
2. ttl (int) – Optionally, one may provide an integer as second parameter in the decrypt method. The ttl denotes the time about how long a token is valid. If the token is older than ttl seconds (from the time it was originally created) an exception is thrown. If ttl is not passed as a parameter, then age of the token is not considered. If the token is somehow invalid, an exception is thrown. Returns value: Returns the original plaintext.

5.7. Data Flow Diagram

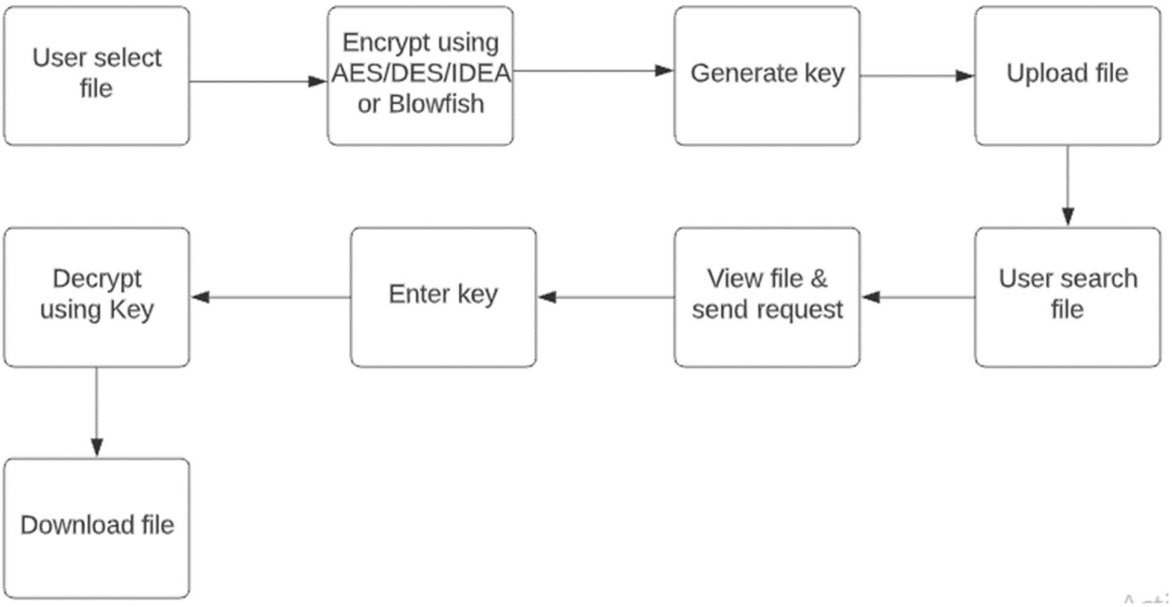


Fig. General Block Diagram of the Project

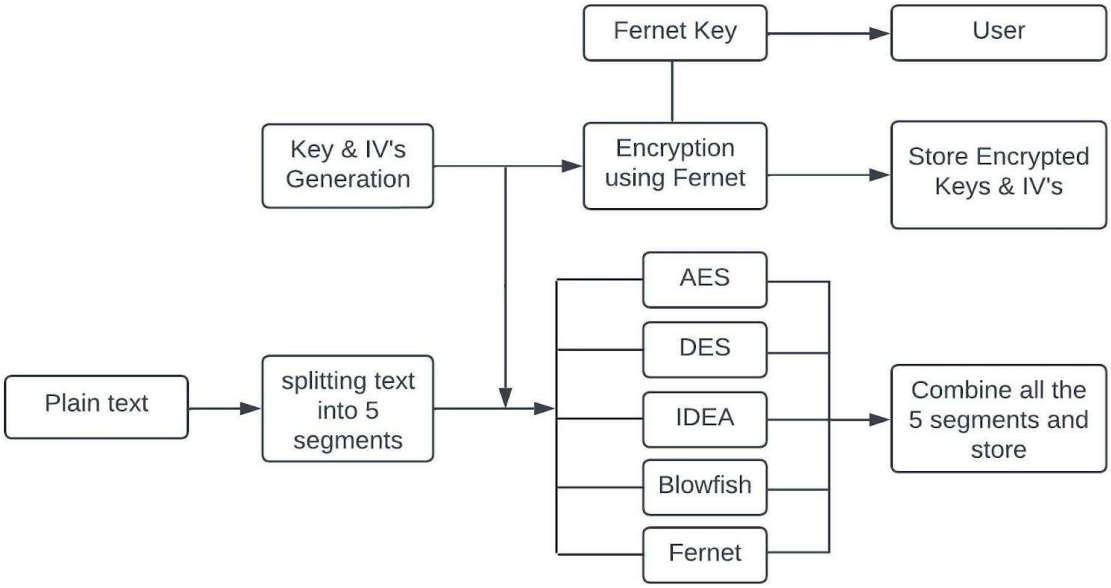


Fig. Block Diagram of project workflow Algorithm

6.Project Implementation

We have implemented our project in 3 stages:

Stage 1: In stage 1 , we created a basic encryption model using AES and DES Hybrid in python. In which a text file was encrypted using AES and DES both in 2 segments and the common key was given to the user.

Stage 2: Then to improve the security and make it more complex to decrypt we implemented a total of 5 Encryption algorithms (AES, DES, IDEA, Blowfish, Fernet). The input text is split into 5 segments and is given as input to 5 Encryption techniques. For encryption the keys are generated along with the IVs using os.urandom() method and the key for fernet is generated using Fernet.generate_key(). These keys and IVs are given as input along with the segments to the encryption algorithms. The output is combined and saved as a single file. The keys are encrypted using fernet and stored and the fernet key used for this is given to the user. Once this model was ready it was tested.

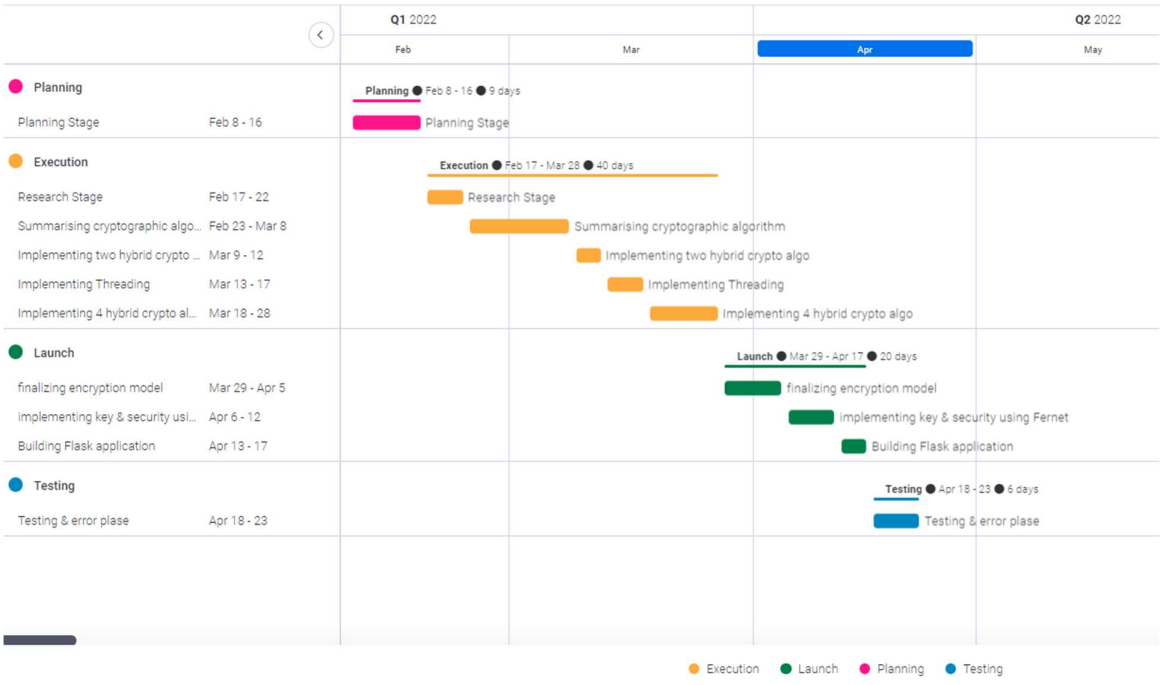
Stage 3: We created a basic web app using Flask for the model. It has the home page where users can upload the file for encryption and encrypt it and download the key. Next page has the decryption option where user can decrypt the file and download it.

7. Result

The user visits the website to encrypt his/her document, after the document is uploaded it goes into the process of splitting and encryption. Encryption is done using the five algorithms mentioned above in a simultaneous manner, after the encryption the user is provided with a key which can be downloaded. If the user wants the document, he/she can use the decrypt function with their respective key and after the decryption process is complete the file can be downloaded.

8. Annexure

8.1.Gantt Chart



9. Bibliography

[1] Pahal R and Kumar V 2013 Efficient Implementation of AES (International Journal of Advanced

[2] William S 2017 Cryptography and network security principles and practice 7th Ed (Prentice-Hall Inc) 23-50.

[3] Hercigonja Z 2016 Comparative Analysis of Cryptographic Algorithms (Croatia: International Journal of Digital Technology & Economy 1 127-34.

[4] Al-Nbhany W, A N A and Zahary A T 2016 A Comparative Study among Cryptographic Algorithms: Blowfish, AES and RSA (Morocco: International Arab Conference on Information Technology (ACIT2016) Beni Mellal).

[5] Tripathi R and Agrawal S 2014 "Comparative Study of Symmetric and Asymmetric

Cryptography Techniques (International Journal of Advance Foundation and Research in Computer (IJAFRC)) 1 68-76.

[6]Ruchita Sharma, Swarnalata Bollava1.Ruchita Sharma, Swarnalata Bollavarapu(2015).Data Security using Compression and Cryptography Techniques. 2015International Journal of Computer Applications.

[7]Mahavir Jain,Arpit Agrawal(2014).Implementation Of Hybrid Cryptography Algorithm. International Journal Of Core Engineering & Management(IJCEM).

[8]Kumar, A., Lee, B. G., Lee, H., & Kumari, A. (2012). Secure storage and access of data in cloud computing. 2012 International Conference on ICT Convergence (ICTC).

rapu(2015).Data Security using Compression and Cryptography Techniques. 2015International Journal of Computer Applications.

[9]Mahavir Jain,Arpit Agrawal(2014).Implementation Of Hybrid Cryptography Algorithm. International Journal Of Core Engineering & Management(IJCEM).

[10]Kumar, A., Lee, B. G., Lee, H., & Kumari, A. (2012). Secure storage and access of data in cloud computing. 2012 International Conference on ICT Convergence (ICTC).