**DATA STORAGE USING HYBRID CRYPTOGRAPHY AND STEGANOGRAPHY**

**A PROJECT REPORT**

***Submitted by***

**VIGNESH G**

**RAGHURAJ S V**

**DHAMODARAN S**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**

**COLLEGE OF ENGINEERING,GUINDY**

**ANNA UNIVERSITY : CHENNAI 600 025**

**JUNE 2022**

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report “**DATA SECURITY USING HYBRID CRYPTOGRAPHY AND STEGANOGRAPHY”** is the bonafide work of “**VIGNESH, RAGHURAJ, DHAMODARAN ”** who carried out the project under my supervision, for the fulfillment of the requirements as part of the CS6611 Creative and Innovative Project Laboratory

**Place:**

**Date:**

**SIGNATURE**

Mr.T.M.Thiyagu

**SUPERVISOR**

Teaching Fellow

Department of Computer Science and Engineering

Anna University,

Chennai-600025

**ABSTRACT**

Data security is the main concern in different types of applications from data storing in clouds to sending messages using chat. In order to provide security for data transfer, there are many types of techniques which are already proposed like AES, DES, RSA ,RC-6 but in existing methods, most of the time only a single type of encryption was used based on user requirement but in this system main problem is each encryption is done using encryption keys if these keys are exposed in any case entire data is lost so we need an effective method which can provide more security so in this project hybrid cryptography is used where existing encryption methods are used but three methods will be used. When the user uploads data will split into three parts and the first part will be encrypted using Blowfish, the second part will be encrypted using AES, the third part will be encrypted using TrippleDES and these three encrypted files and keys used for AES, TrippleDES and Blowfish are combined using AES encryption cryptography algorithm and stored in the image using LSB steganography when users want to download total data first keys should be retrieved from the image and these keys are used for decrypting data again by using AES, TrippleDES and Blowfish and final data is combined and stored in the file. This method provides more security for data.

**TABLE OF CONTENTS**

**Chapter No: TITLE PAGE No:**

**ABSTRACT**

**LIST OF TABLES**

**LIST OF FIGURES**

**LIST OF SYMBOLS**

**LIST OF ABBREVIATION**

**1. INTRODUCTION**

1.1 Objective

1.2 Problem Statement

1.3 Proposed Solution

1.4 Need For the System

1.5 System Analysis

**2. LITERATURE SURVEY**

2.1 Literature of Related Works

2.2 Issues Identified

2.3 Summary of Related Works

**3. SYSTEM DESIGN**

3.1 System Architecture

3.2 System Specification

3.2.1 Hardware Specification

3.2.2 Software Specification

**4. DETAILED ARCHIETECTURE**

4.1 List of Modules

4.1.1 Data Preprocessing

4.1.2 cryptography Algorithm

4.1.3 Steganography and key Transfer

4.1.4 Decryption and Extraction

**5. IMPLEMENTATION AND RESULTS**

5.1 Data Preprocessing

5.2 Cryptography Algorithm

5.3 Steganography and Key Transfer

5.4 Decryption and Extraction

**6. TEST CASES AND PERFORMANCE METRICS**

6.1 TEST CASES

6.2 PERFORMANCE METRICS

**7. CONCLUSION AND FUTURE WORK**

7.1 CONCLUSION

7.2 FUTURE ENCHANCEMENT

**REFERENCES**

**LIST OF TABLES**

**Table.No Description PAGE.NO:**

6.1 Testcases

6.2 Performance Metrics

**LIST OF FIGURES**

**Fig.No: Description PAGE.NO.**

1.5.1 AES Block diagram

1.5.2 Blowfish Block Diagram

1.5.3 TripleDES Block Diagram

3.1 System Architecture

4.1.1 Module-1 Block Diagram

4.1.2 Module-2 Block Diagram

4.1.3 Module-3 Block Diagram

4.1.4 Module-4 Block Diagram

**LIST OF SYMBOLS**

**^ -** append string

**⊕ -** Logical XOR function

**LIST OF ABBREVIATION**

AES- Advanced Encryption Standard

DES- Data Encryption Standard

LSB- Least Significant Bit

IV-Initialization Vectors

CBC-Cipher Block Chaining Mode

SHA- Secure Hash Algorithm

**CHAPTER - 1**

1. **INTRODUCTION**

The widen handling of digital media for information transmission through secure and unsecured channels exposes messages sent via networks to intruders or third parties. Encryption of messages in this modern age of technology becomes necessary for ensuring that data sent via communications channels become protected and made difficult for deciphering. Enormous number of transfer of data and information takes place through internet, which is considered to be most efficient though it’s definitely a public access medium.Therefore to counterpart this weakness, many researchers have come up with efficient algorithms to encrypt this information from plain text into ciphers. In information security, encryption is the process of transforming information using an algorithm to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. The result of the process is encrypted information. The reverse process is referred to as decryption. Single cryptosystems are now a days cracked with latest hardware capabilities so a hybrid system is proposed in this paper.

**1.1Objective**

The main objective of this project is to build a Hybrid Crypto-system that secures data on multiple layers and also ensures security of keys. The Hybrid cryptosystem also securely stores the keys so that they don’t lead to any vulnerabilities. To create a crypto-system that provides excellent security without compromising on performance and speed. To overcome the performance-security tradeoffs of cryptographic algorithms when used separately.

**1.2 Problem Statement**

Various Encryption Algorithms are used in apps and services to secure data. But the advent of new and sophisticated technologies is making these existing systems obsolete. Advancements in Hardware have significantly reduced the time required to break a cryptographic system. Various kinds of attacks have weakened the existing systems.

Crypto-analysis and special mathematical attacks have made these systems quite vulnerable to being broken by cryptographers. Key security is another vulnerability that modern systems face. Ensuring safe storage and transmission of sensitive keys is a major fault of existing systems. Another key aspect of securing data is to ensure that performance is not compromised.

Generally, encryption algorithms to provide higher levels of security use larger key lengths, but that hampers the performance of the system. A single layered standalone crypto-system can sometimes have trade-offs that might lead to data leaks, and also hamper key security. A standalone system has vulnerabilities that often effect the security of data. The various pitfalls of standalone systems at times compromise the performance and speed.

**1.3 PROPOSED SOLUTION**

The solution is to secure the data for storage and transmission by building an application that takes the confidential data as input and undergo splitting of data , involve in hybrid cryptography algorithm[Blowfish , AES, TrippleDes] hide the encrypted the keys in an image using LSB steganography , and decryption application decrypts the data from image and encrypted files.

Since we are splitting the data and encrypting them the hackers find it difficult apply which encryption algorithm to which data, even though if he decrypts it only a part of the data can be accessed. So, it makes brute force attacks normal time n into n power 3. The keys are encrypted using the hash of the password, as the key for AES. SHA-1 is used to generate the hash from the user input password and stored in an image using LSB steganography making it unnoticeable to the hackers

**1.4 Need For the System**

The file is being encrypted by using symmetric key cryptography and stenography techniques. The system is very secure and robust. Data of the users is secured on a by our hybrid cryptography algorithms and steganography which helps in avoiding unauthorized access from the outside world.. Data security is a major priority. This system can be implemented in the banking and corporate sectors to securely transfer confidential data.

**1.5 System analysis**

**1.5.1 Cryptography**

Cryptography is the study of techniques and methods of securing data and communication between different parties, from malicious intruders, eavesdroppers, and adversaries. Cryptography involves various aspects of securing data communication like creating protocols to facilitate secured and safe connectivity, analyzing cryptosystems for vulnerabilities, and performance and checking for the fulfillment of information security aspects like integrity, authenticity, confidentiality, and non-repudiation.

Cryptography is the backbone of Information security and Information Security is essential for a user’s digital presence. Cryptography is primarily associated with Encryption. Encryption is the procedure of converting readable information/data to unreadable gibberish. Encryption is facilitated by the use of Cryptographic Algorithms or Cryptosystems. Modern cryptosystems are heavily dependent on mathematical theories and functions as well as theoretical computer science practices. Encryption is facilitated using mathematical functions that require a user-known secret, called the Key.

Decryption reverses what encryption does, that is, decryption converts the unintelligible gibberish back into readable information. Using the Key. Cryptography algorithms can be classified into two categories - Asymmetric Cryptography and Symmetric-Key Cryptography. Symmetric-Key algorithms for Cryptography uses the identical keys for both encryption, as well as for decryption, whereas Asymmetric Cryptography uses different keys.

**1.5.2 Advacnced-Encryption Standard(AES)**

Rijndael, proposed by Belgian cryptographers, Vincent Rijmen and Joan Daemen, is Symetric-key Block-Cipher that has been established as the Advanced Encryption Standard by the National Institute of Standards and Technology (NIST) of The United States of America, in the year of 2001.

The AES may have keys varying in size between 128, 192, 256 bits, & having 10, 12, 14 rounds respectively. The n-Bits key is expanded using AES Key-Scheduling into several subkeys depending on the number of rounds. In the beginning, the input block is XORed with an Initial Round-Key. Then, for the first N-1 rounds, 4 Round Functions are applied on each block. The first-round function is Substitute Bytes where every byte is substituted by another, from the lookup table. Followed by Shift-Rows, where the last three rows are cyclically shifted by certain number of steps. Shift-Rows is followed by Mix-Columns, where a linear mixing operation is executed on the columns, combining the 4-bytes of each column. Lastly, Add-RoundKey function is executed on the current state, where each byte, and a byte of the round key are combined using bit-wise XOR operation. For the Nth round, i.e the last round all the above functions are applied except the Mixed Columns step.

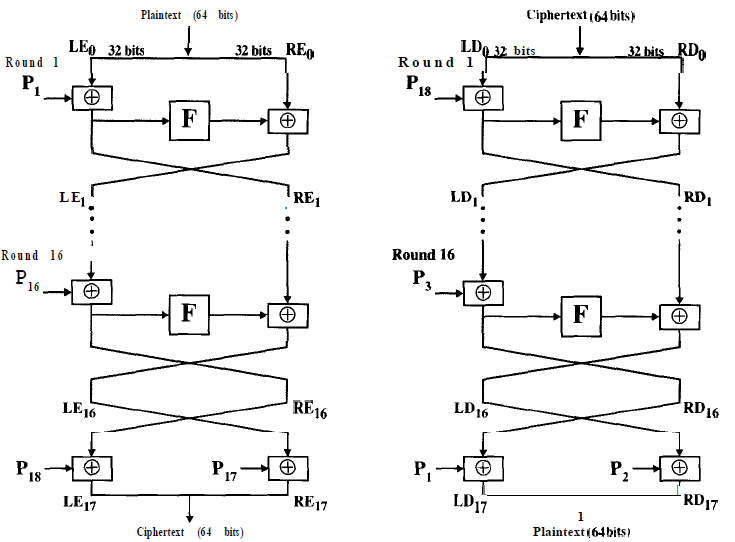
AES is one of the most extensively used and secure algorithms for data security. Even though it is slower than blowfish, it provides a higher level of data security.

****

**Fig.No.1.5.1** AES block diagram

**1.5.3 Blowfish Algorithm**

Blowfish, a Symmetric-Key Block Cipher, was developed by B. Schneier in the year 1993. Blowfish algorithm has 64 Bits block size and variable key length of 32 to 448 Bits. It is particularly known for its features like complicated key schedules and keydependent s-boxes. Being a Feistel cipher it has 16 rounds. Each round, consists of four steps. In nth round, the left half of the block and the nth element in the subkey-array are XORed  **⊕** followed by passing it to the round function F. The return from the function F and the right half of the initial block are XORed  **⊕** and then swapped. The round function F divides the 32-bit input into four 8-bit blocks that are then fed to 4 different S-Boxes. The returns from the 1 st and 2 nd s-box are added and the return is XORed  **⊕** with the returns from the 3 rd s-box and again added with the output of the 4 th s-box. It is one of the fastest algorithms for encryption.

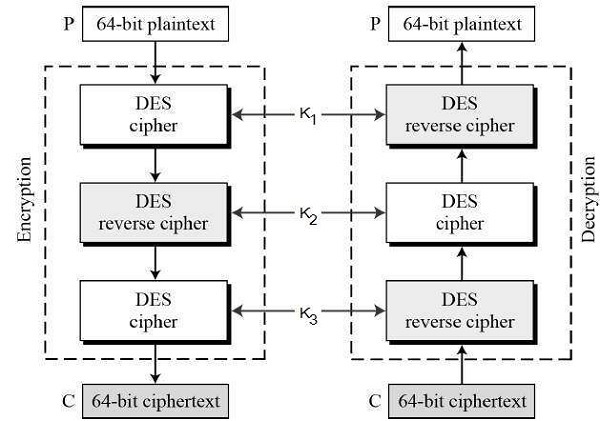


**Fig 1.5.2** Blowfish Block Diagram

**1.5.4 Triple-DES Encryption**

The DES (Data Encryption Standard) In cryptography, Triple DES (3DES or TDES),officially the Triple Data Encryption Algorithm (TDEA or Triple DEA), is a symmetric-keyblock cipher, which applies the DES cipher algorithm three times to each data block. Whilethe government and industry standards abbreviate the algorithm's name as TDES (TripleDES) and TDEA (Triple Data Encryption Algorithm), RFC 1851 referred to it as 3DES fromthe time it first promulgated the idea, and this namesake has since come into wide use bymost vendors, users, and cryptographers. Before using 3TDES, user first generate anddistribute a 3TDES key K, which consists of three different DES keys K1, K2 and K3. This means that the actual 3TDES key has length 3×56 = 168 bits.

* Encrypt the plaintext blocks using single DES with key K1.
* Now decrypt the output of step 1 using single DES with key K2.
* Finally, encrypt the output of step 2 using single DES with key K3.
* The output of step 3 is the cipher text



**Fig.No:1.5.3** TripleDES Block Diagram

**1.5.4 LSB-Image Steganography**

Least Significant Bit Steganography is a technique of hiding data within digital media, here, Image. Images are made up of pixels, and the value of each pixel usually refers to the color-code of that pixel. In a photo’s gray-scale mode, these pixel values range from 0-255. In LSB Image Steganography, the least-significant bit of a pixel is changed, but that doesn’t have much of a visible change in the image. A cover image is where the data is hidden. The cover image is converted to grey scale. The message is converted into binary. Each pixel of the image is traversed through, and for each pixel, initiate a temporary variable, temp. If the LSB of the Pixel Value and the message bit is the same, set temp as 0 and set temp as 1 otherwise. Update the output image pixel as image pixel value added with the temporary variable value, temp. This is done until the message is completely embedded.

**1.5.5 SHA-1 HASHING FUNCTION**

Secure Hashing Algorithm is a one-way hash function that generates a condensed hash of the message, called the message-digest. Any changes made to the message get reflected onto the message-digest, that is if the message changes the message digest will change. This feature of SHA-1 is highly efficient in the generation of random numbers and bits, generation and validation of digital signatures, message authentication codes.

**CHAPTER - 2**

**2. LITERATURE SURVEY**

**2.1 Literature of Related Works**

**2.1.1** Authors, Publication & Year: Chitra Biswas, Udayan Das Gupta, Md. Mokammel Haque - ICECCE, 2019

The resistivity of the system proposed by Biswas et al. (2019) , consisting of AESRSA Data and Key security and LSB Steganography for storing encrypted key, against attacks has been eshtablished. Thus this system provides authentication, integrity and confidentiality together.

# 2.1.2 Authors, Publication & Year:[Putta Bharathi](https://ieeexplore.ieee.org/author/37088926720), [Gayathri Annam](https://ieeexplore.ieee.org/author/37088926633), [Jaya Bindu Kandi](https://ieeexplore.ieee.org/author/37088925843), [Vamsi Krishna Duggana](https://ieeexplore.ieee.org/author/37088926482) -ICECCE, 2021

The is encrypted using AES,DES and RSA.The key is securely stored in an image using the LSB algorithm. This ensures that the file is completely secured.

**2.1.3** Authors, Publication & Year: Muhammad Abdul Muin, Muhammad Abdul Muin, Arief Setyanto, Sudarmawan, Kartika Imam Santoso - ICITACEE, 2018

The conclusion of the research by Muin et al. (2018) shows that a hybrid cryptosystem on AES256-Blowfish required longer decryption time compared to the one in reverse order (Blowfish-AES256) therefore, is proven to be the most secured one when compared with Blowfish-AES256, Blowfish, or AES256.

**2.1.4** Authors, Publication & Year: Ye Liu, Wei Gong, Wenqing Fan - ICIS, 2018

They showed that Combining asymmetric encryption with symmetric encryption algorithms makes the system significantly secured and faster. The experimental system also shows that Hybrid Crypto-systems are a great alternative to traditional crypto-systems that rely on higher keys sizes and rounds.

**2.1.5** Authors, Publication & Year: Prof. Swapnil Chaudhari, Mangesh Pahade, Sahil Bhat, Chetan Jadhav, Tejaswini Sawant - IJRDT, 2019

It studies the implementation of a hybrid Crypto-system of symmetric encryption and public-key algorithms. The paper also explores the flows in both standalone systems and uses a hybrid approach to enhance security and address the drawbacks of the standalone systems

**2.1.6** Authors, Publication & Year: Dr. Mahmood Zaki Abdullah, Zinah Jamal Khaleefah

Transactions on Image Processing (Journal), 2018 Abdullah and Khaleefah (2018) proposes a system that establishes the use of hybrid crypto-systems increases the level of security of encrypted data and also reduces the time required for encrypting data and decrypting ciphertext.

**2.2 Issues Identified**

Splitting the given information into 3 parts and applying both symmetric and Asymmetric algorithms lead to confusion and time consuming so we are using only symmetric encryption algorithms. Placing the encrypted file in cloud makes on dependent on internet facility and cloud server availability. There are combinations of cryptography algorithm that can be applied but some combinations tend to time consuming and less secure.

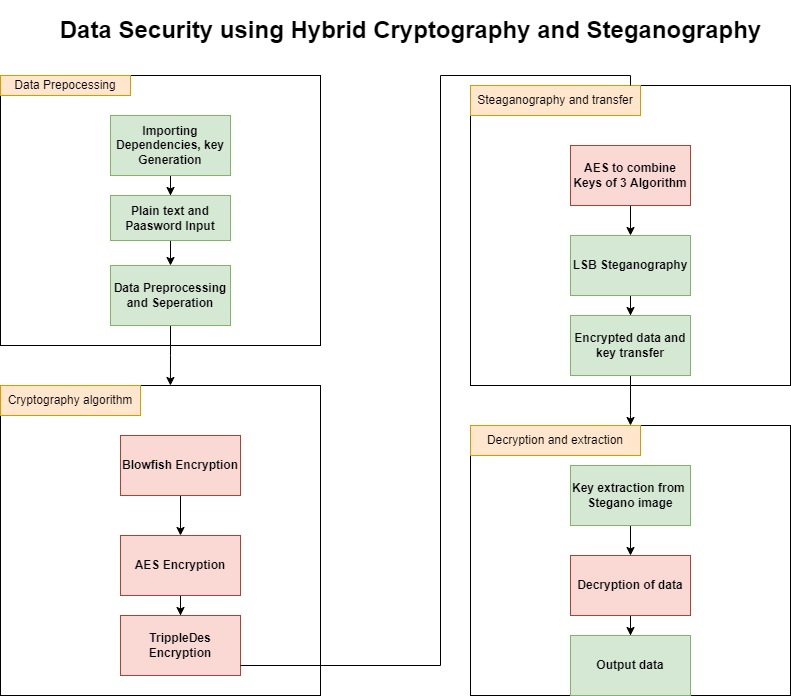
**2.3 Summary of Related works**

In the proposed system, the main goal is to Encrypt data using Cryptographic Algorithms like Blowfish, AES and TrippleDes in a cascading manner using the least resources and with least complexity. The use of multiple encryptions makes the data even more secure without the requirement of Higher Key lengths. The System consists of three Encryption Layers, a Key Generator and a List of Keys. The Key generates the random n-bits Key depending on the Encryption Algorithm, while the List of Keys stores the Key Generated in each layer. From the List of Keys, an AES Encryption Block encrypts it and a LSB Steganography Block to embed the Keys into a Cover Image. And send to the receiver via mail, where he uses decryption process to obtain the confidential data.

**CHAPTER - 3**

**SYSTEM DESIGN**

**3.1 System Architecture**

****

**Fig.No.3.1 System architecture diagram**

**3.2 System Requirements**

**3.2.1 Hardware Specification**

The System doesn’t require any specialized hardware as it doesn’t carry out a large amount of processing.

The basic requirements are:

• **Processors**: Intel’s Atom® processor or Intel’s Core™ processor i3 or above

• **Disk space**: Recommended disk space is 1 GB

• **RAM**: 2GB or more

**3.2.2 Software Specification**

The project requires an operating system and other software needed for the execution of this application. Operating System provides the underlying environment for the execution of the application. Additional software is needed for the development.

• **Operating systems**: Microsoft Windows 7 or above, Apple’s macOS, & Linux

• **Python versions**: Python 3.6.X and above

• **Libraries**: pycrypto, stegano, random and other general purpose libraries

The project is implemented in Python with the help of a few libraries as mentioned next.

**3.2.2.1 Python Libraries**

The Python Libraries used for the project are:

* **PyCrypto:** It contains several containers and functions in Asymmetric and symmetric cryptography algorithms and hash functions.
* **Hashlib:** It contains multiple hash functions and operations. The constructors in the library provide some of the hashing functions used in this project.
* **Stegano:** This library is used to facilitate steganography in the project.
* **Random:** The random library is used to generate random numbers and strings for encryption.

**CHAPTER - 4**

**DETAILED ARCHITECTURE**

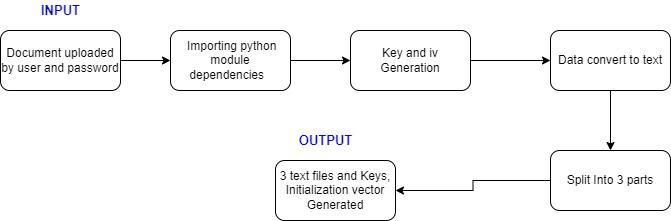
**4.1 List of Modules**

1. Data Preprocessing
2. Cryptography Algorithm
3. Steganography and Key Transfer

4) Decryption and Extraction

**4.1.1 Data Preprocessing**

Data to be sent in encrypted manner is received from the user along the key/password going to be used to encryption. The data given the user in any format such as pdf or word is converted to a text sequence enabling us to make encryption. Key generation block is used to generate random keys for the cryptography algorithm , which is generally 16 bit long.



**Fig.No.4.1.1** Module-1 Block Diagram

**Input:** File to encrypted, password.

**Output:** Processed text sequence and Keys for cryptography algorithm.

**Pseudo code:**

**Key generation:**

def key\_generator(size, case="default", punctuations="required"):

    if case=="default" and punctuations=="required":

        return ''.join(random.choices(string.ascii\_uppercase + string.ascii\_lowercase + string.digits + string.punctuation, k = size))

    elif case=="upper-case-only" and punctuations=="required":

        return ''.join(random.choices(string.ascii\_uppercase + string.digits + string.punctuation, k = size))

**Document segmentation:**

for i in range(0,3):

      name=str(i+1)+".txt"

      f=open(name,'w')

      ctr=0

      for j in range(k,count):

          k+=1

          f.write(con[j])

          ctr+=1

          if(ctr==limit and i!=2):

              f.close()

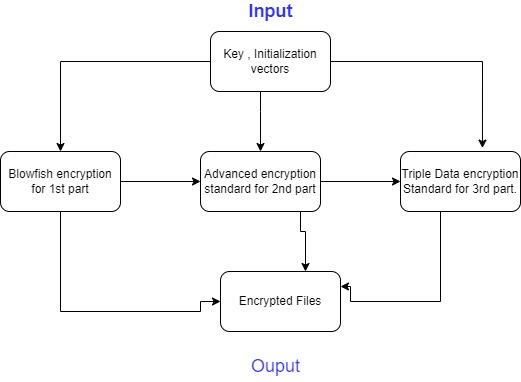
              break

      f.close()

Segment()

**4.1.2 Cryptography Algorithm**

Three Cryptography algorithm such as blow fish, AES, Triple DES are implemented in this module in a for each part of the document separately.First the text sequence is applied to BlowFish algorithm along with the generated key and Initialization vector to obtain a cipher text. Similarly AES and TripleDES applied along with their keys and IV’s for 2nd and 3rd part of text document and corresponding Encrypted files are obtained. All algorithms use CBC mode since plain text size is large than that what a encryption algorithm generally accepts.



**Fig.No.4.1.2** Module-2 Block diagram

**Input:** Text sequence of 3 parts, keys and Initialization vectors

**Output**: Encrypted files-cipher text.

**Pseudo code:**

**Blowfish:**

cipher = Cipher(algorithms.Blowfish(key), modes.CBC(iv), backend=backend)

encryptor = cipher.encryptor()

 cont = encryptor.update(content) + encryptor.finalize()

 open(os.path.join("1.txt"),"w").close()

 f=open(os.path.join("1e.txt"),"wb")

 f.write(cont)

 f.close()

**AES**

cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=backend)

encryptor = cipher.encryptor()

cont = encryptor.update(content) + encryptor.finalize()

 open("2.txt","wb").close()

 f=open("2e.txt","wb")

 f.write(cont)

 f.close()

**TripleDES**

cipher = Cipher(algorithms.TripleDES(key), modes.CBC(iv), backend=backend);

encryptor = cipher.encryptor();

cont = encryptor.update(content) + encryptor.finalize();

open("3.txt","w").close();

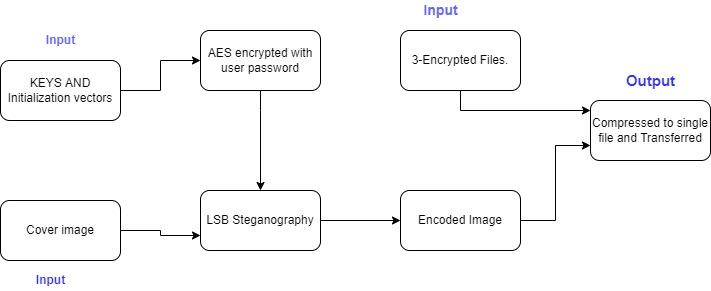
 f=open("3e.txt","wb");

f.write(cont);

 f.close();

**4.1.3 Steganography and key Transfer**

Least significant bit steganography technique is used to with a selected cover image. The keys and initialization vectors used in crypto-algorithms are combined using AES algorithm by password obtained from the user and hidden in the image using a cover image PNG file The stegano-image where the keys are hidden in the least significant bit of the image along with the encrypted file is compressed into a single file send to the receiver in email.



**Fig.No.4.1.3** Module-3 Block diagram

**Input:** Encrypted files, keys,IV vector,Image.

**Output:** Compressed ZIP file-Enc files and encoded image.

**Pseudo code:**

**AES Encryption of keys**

hash = hashlib.sha1()

hash.update(password.encode())

password\_encryption\_cipher = AES.new( hash.hexdigest()[:16].encode() , AES.MODE\_CBC, iv= '16bitAESInitVect'.encode())

encrypted\_keys\_and\_iv=hexlify(password\_encryption\_cipher.encrypt(pad(json.dumps(keys\_iv).encode(), AES.block\_size)))

**LSB Steganography:**

  r, g, b = to\_binary(pixel)

if data\_index < data\_len:

                pixel[0] = int(r[:-1] + binary\_secret\_data[data\_index], 2)

                data\_index += 1

            if data\_index < data\_len:

                pixel[1] = int(g[:-1] + binary\_secret\_data[data\_index], 2)

                data\_index += 1

            if data\_index < data\_len:

                pixel[2] = int(b[:-1] + binary\_secret\_data[data\_index], 2)

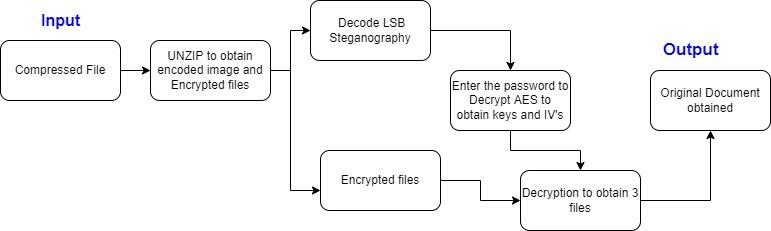
                data\_index += 1

            if data\_index >= data\_len:

                break

**4.1.4 Decryption and Extraction**

The compressed File is unzipped to obtain encoded stegano image and Encrypted files. The Stegano-image received is decrypted using AES Decryption with the password to obtain the combined keys. The keys are separated and given to encrypted files of Blowfish,AES AND TripleDES separately to decrypt the file into 3 parts. The Decrypted files are merged to Obtain the original document.



**Fig.No.4.1.4** Module-4 Block diagram

**Input:** Compressed File-containing encoded image and encrypted file.

**Output:**  decrypted original file

**Pseudo code:**

**Decode from Stegano-image:**

for row in image:

        for pixel in row:

            r, g, b = to\_bin(pixel)

            binary\_data += r[-1]

            binary\_data += g[-1]

            binary\_data += b[-1]

     all\_bytes = [ binary\_data[i: i+8] for i in range(0, len(binary\_data), 8)

    decoded\_data = ""

    for byte in all\_bytes:

        decoded\_data += chr(int(byte, 2))

        if decoded\_data[-5:] == "=====":

            break

**AES Decryption of keys:**

hash = hashlib.sha1()

hash.update(password.encode())

password\_decryption\_cipher = AES.new( hash.hexdigest()[:16].encode() , AES.MODE\_CBC, iv= '16bitAESInitVect'.encode())

decrypted\_keys\_iv=json.loads(unpad(password\_decryption\_cipher.decrypt(unhexlify(unhide\_encrypted\_keys\_and\_iv)), AES.block\_size))

**Decryption from Blowfish,AES,TripleDES:**

cipher = Cipher(algorithms.Blowfish(key), modes.CBC(iv), backend=backend)

decryptor = cipher.decryptor()

 content=decryptor.update(content) + decryptor.finalize()

 f=open("1d.txt","wb")

 f.write(content)

 cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=backend)

 decryptor = cipher.decryptor()

 content=decryptor.update(content) + decryptor.finalize()

  f=open("2d.txt","wb")

  f.write(content)

cipher = Cipher(algorithms.TripleDES(key), modes.CBC(iv), backend=backend)

 decryptor = cipher.decryptor()

 content=decryptor.update(content) + decryptor.finalize()

 f=open("3d.txt","wb")

 f.write(content)

**Chapter -5**

**IMPLEMENTATION AND RESULTS**

**5.1 Data Preprocessing**

Random function is used to generate keys for blowfish, AES and TrippleDes cryptography algorithm. It generates different set of keys of required length everytime.

**5.1.1 Keys for 3 Algorithms**:

b'?\\9),-\_<I{V/9,l@'

b'%D@UMSj\\4X+L;v0l.!'

b'?zfzpS:2PY0o4rCxn\\z^St79'

**5.1.2 Initialization vectors generated:**

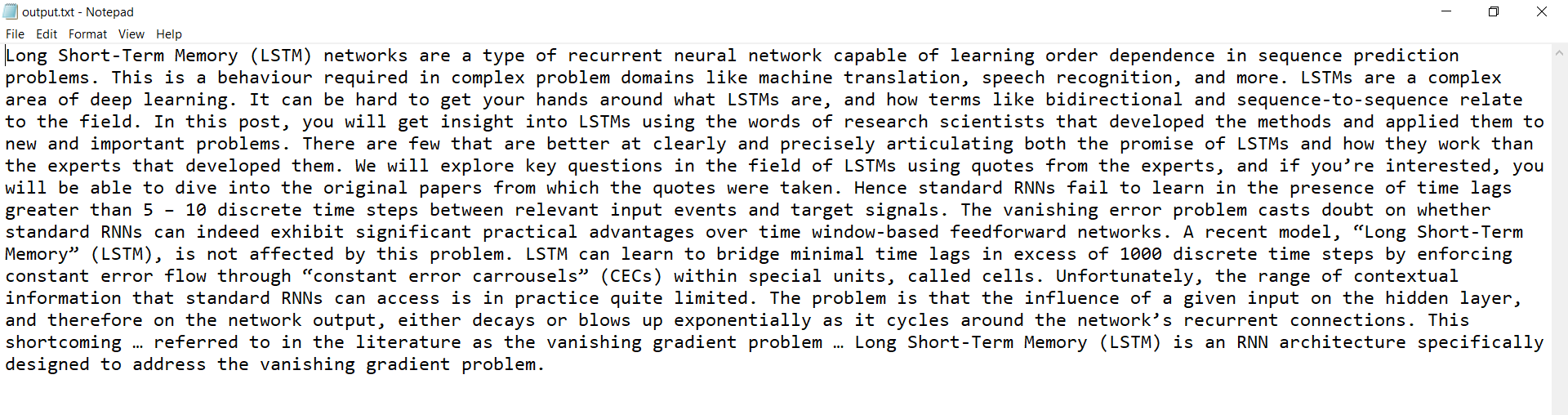
b'\xe1\xf2z\xb0O\xfcr\x10\x81tI\xc0\xafK\xc1F'

b'\xc4\xc2\x1d\x1cj\xddAd'

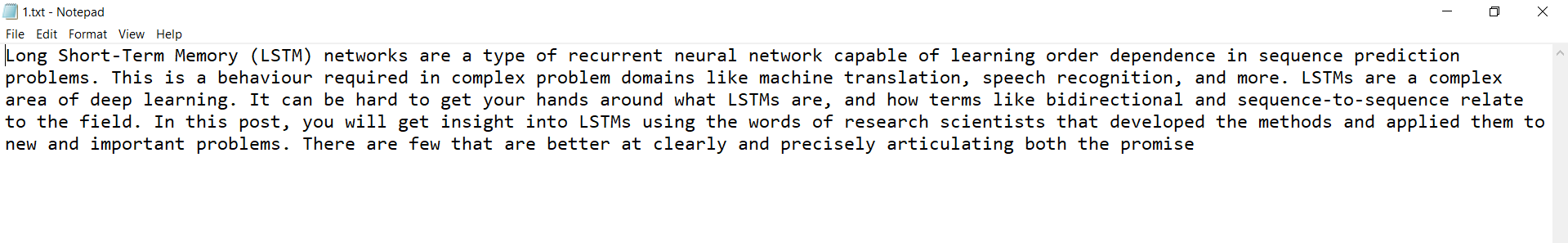
**5.1.3Text document segmentation**:

In order to perform hybrid cryptography (BLOWFISH, AES, TrippleDes ) we have split the document into 3 parts based on the character count. Below are the original document and the 3 parts namely “1.txt, 2.txt, 3.txt”.

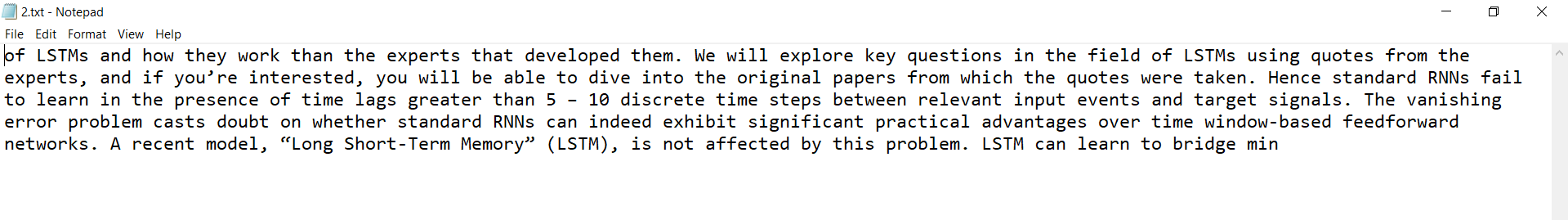
**Original Text file:**

****

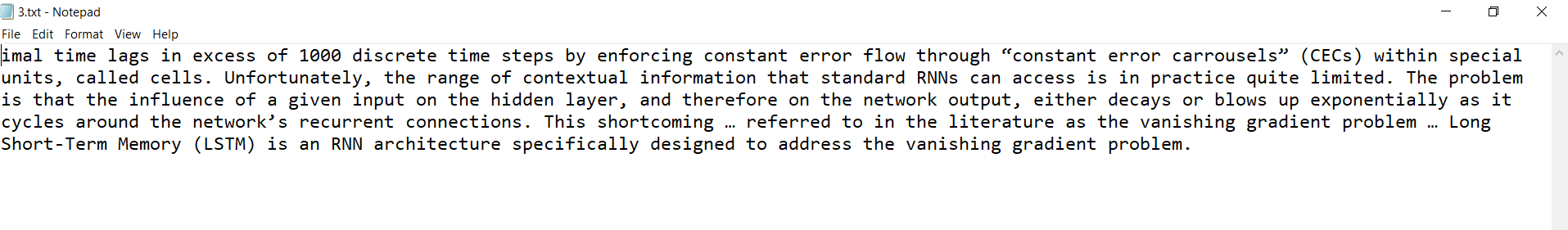
**Segmented File-1:**

****

**Segmented File-2 :**

****

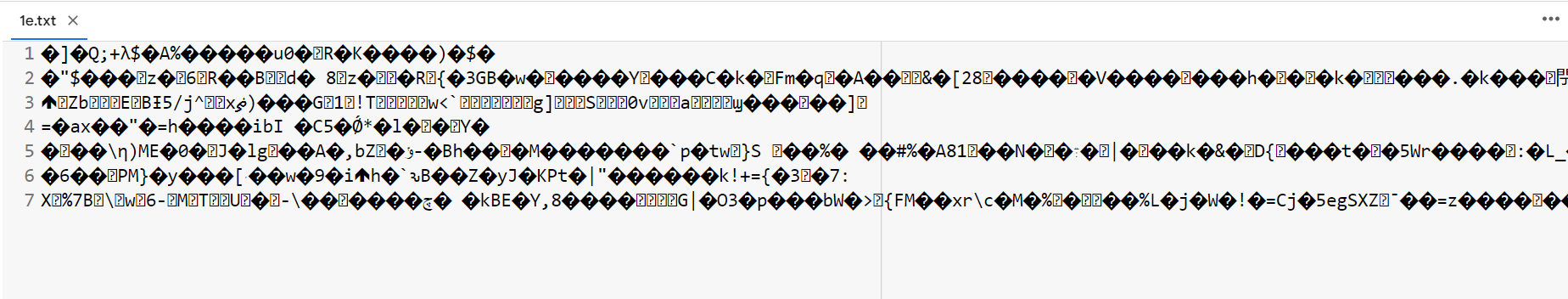
**Segmented File-3:**

****

**5.2 Cryptography Algorithm**

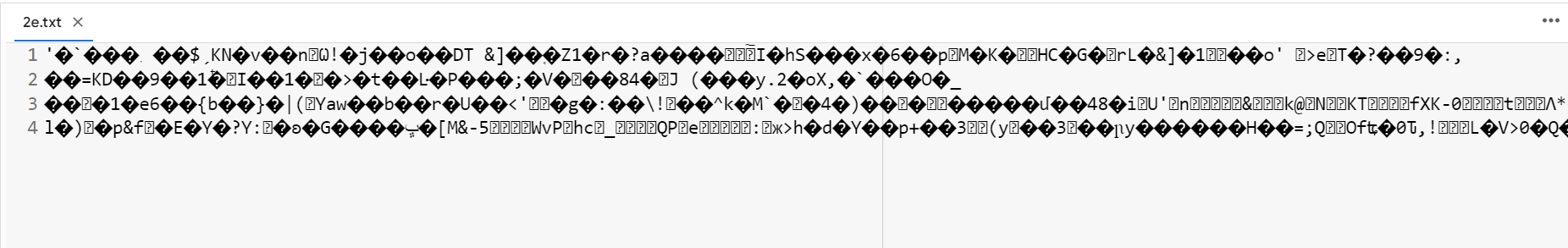
**5.2.1 Blowfish Encryption:**

**1.txt** is encrypted to “1e.txt” file using blowfish. Keys are generated using key generator module.



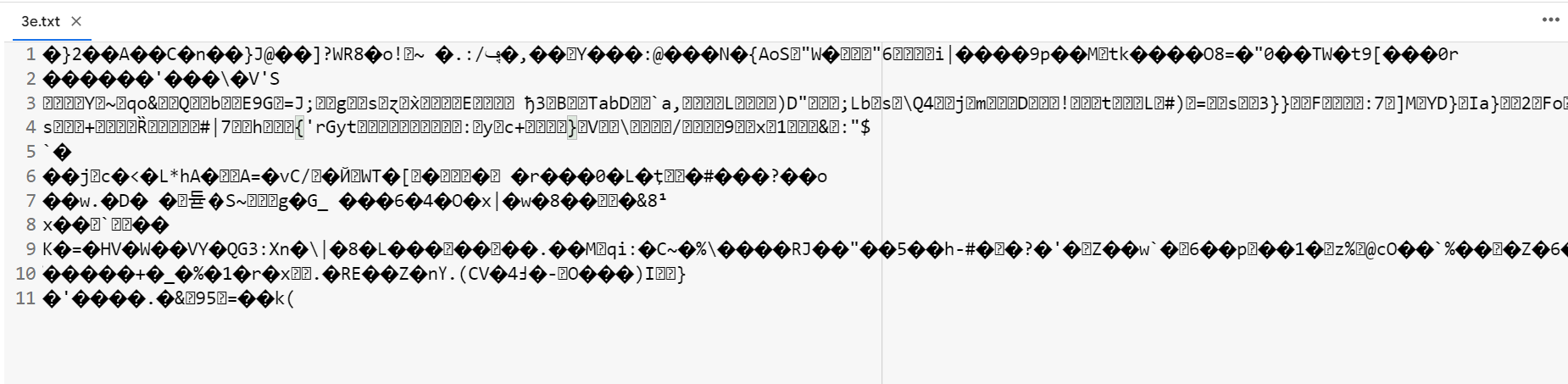
**5.2.2 AES Encryption:**

Second part “2.txt” is encrypted using Advanced encryption standard cryptography algorithm into “2e.txt”.

****

**5.2.3 TripleDES Encryption**

Third part “3.txt” is encrypted using Triple Data encryption Standard or Triple DES algorithm to get encrypted file “3e.txt”.



**5.3 Steganography and Key Transfer**

**5.3.1 AES to combine 3keys**

The Keys used for encryption at the various layers can be securely stored. The List of keys, L stores all the keys generated throughout the Data Encryption Process. Whenever the key for a particular Encryption Layer is generated, it is appended to the List of Keys, L.

In the system, the encryption layers are Blowfish, Triple-Des and AES, respectively, so the Keys used, are stored in the same order as: List of Keys,

L = [ KBlowfish ^ blowfish-iv ^ K-Tripledes ^ TripleDES-iv ^ KAES ^ aes\_iv ]

This List, L is then passed into a function that converts the list into a single string of keys separated by separators ( x , \* , / )

LS = hexlify( L, separator = 0× 0 )

The String, LSB is then encrypted using the AES Encryption Algorithm with a Key generated from user-input password. The user inputs a password, PW which is hashed using SHA1, & the first 16 Bits of the Hash is used as the key KPassword .

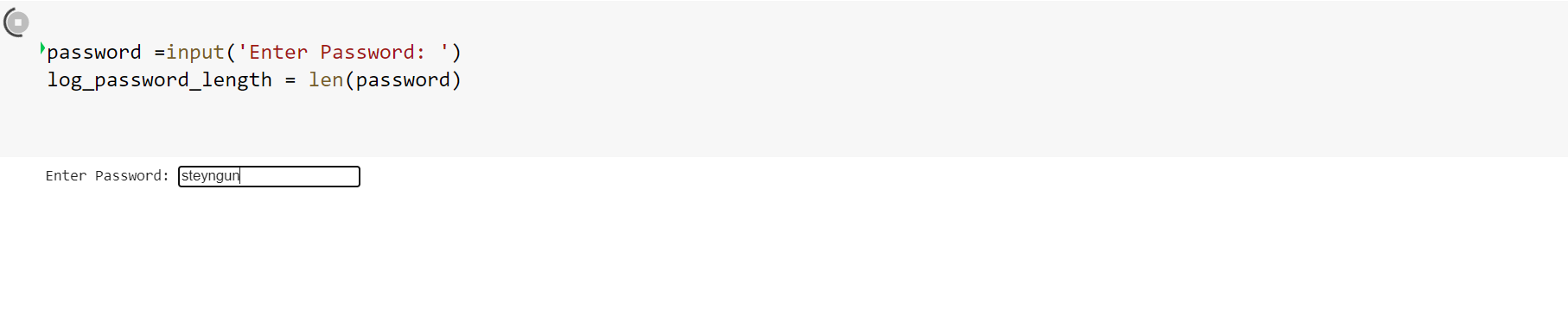
The Key, KPassword is used for the Encryption, generating the encrypted string LS−Encrypted.

HashedPassword, HP = SHA(PW ) Key,

KPassword = HP [0 : 16]

LS−Encrypted = AES(LS, KPassword)

It prompts to enter a password to encrypt the keys using AES.

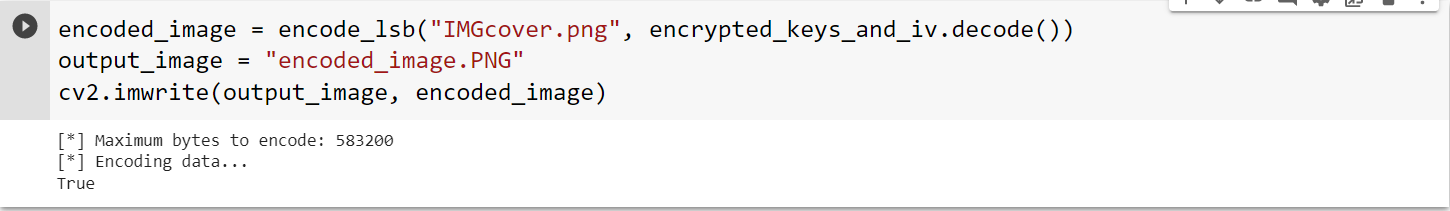


Keys encrypted by AES,for authentication SHA is used.



**5.3.2 LSB Steganography**

The keys encrypted are converted to bits so that they can be placed in lsb of the image.

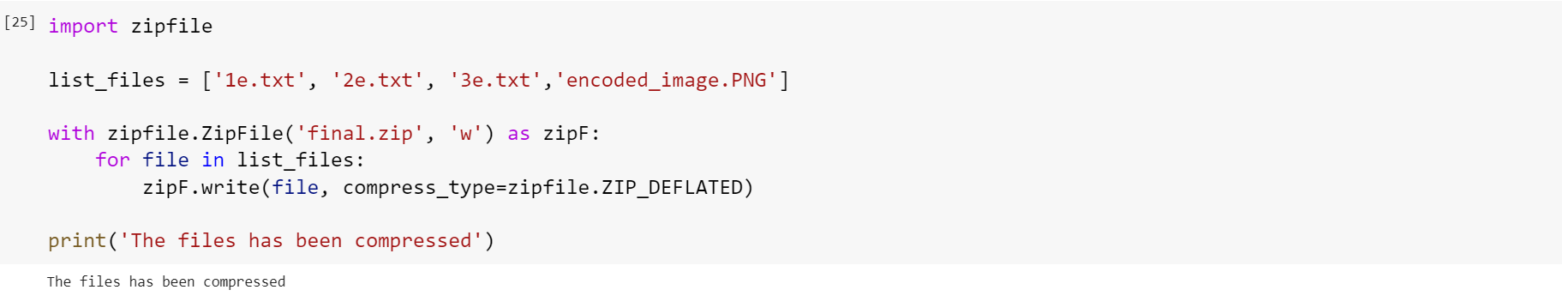


Cover Image Encoded Image

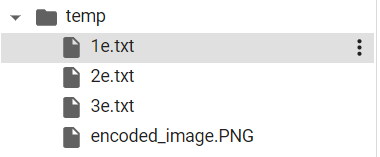
 

**5.3.3 Encrypted data and Key Transfer**

In order to transfer the encrypted files and steganography image,the files are compressed into a zip folder. Named **Final.zip** file. Later the receiver unzips the compressed file into a folder called temp where he gets 1e.txt, 2e.txt, 3e.txt and the encoded image

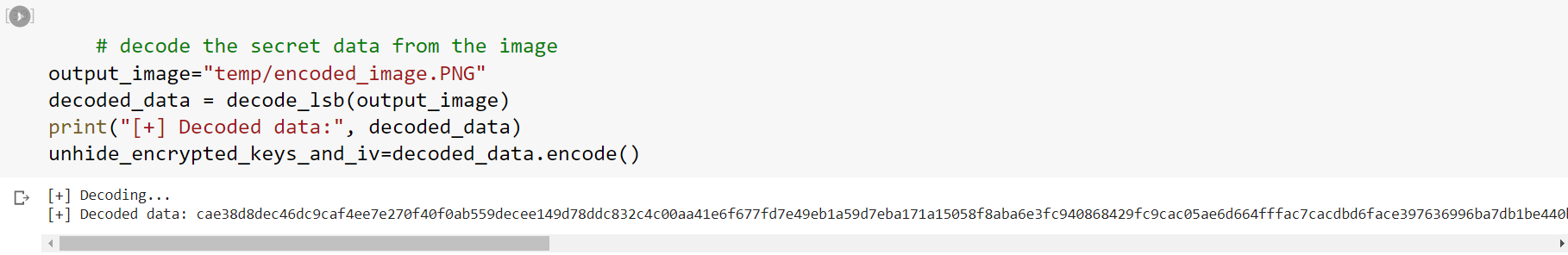


Zip file contains

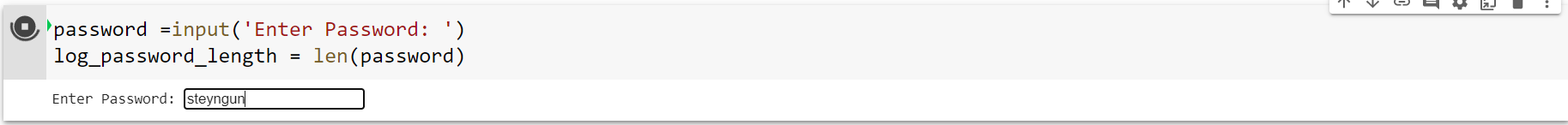


**5.4 Decryption and Extraction of Data**

**5.4.1 Key Extraction from Stegano encoded Image**



We have obtained the keys from the encoded image using reverse of LSB steganography. The have been encrypted by us using AES encryption using a password.(say steyngun here)



We perform AES Decryption and extract our keys used for decrypt our data files.



We have have obtained the keys and Initialization vector used in for encryption, since its symmetric cryptography algorithm same are used to decrypt the data.

**Keys for 3 Algorithms obtained**:

b'?\\9),-\_<I{V/9,l@'

b'%D@UMSj\\4X+L;v0l.!'

b'?zfzpS:2PY0o4rCxn\\z^St79'

**Initialization vectors Obtained:**

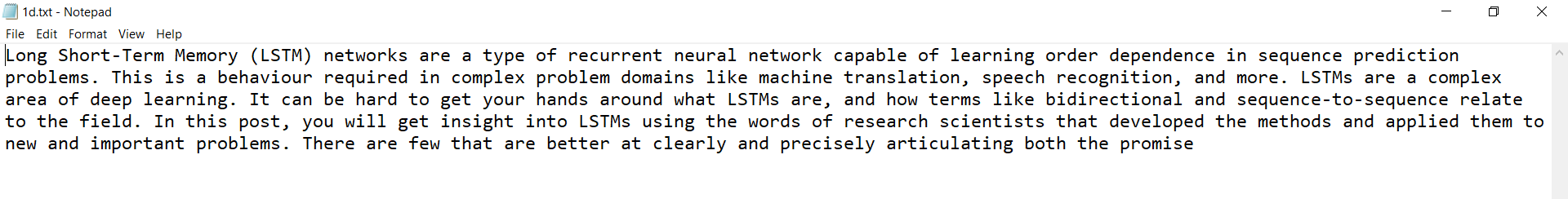
b'\xe1\xf2z\xb0O\xfcr\x10\x81tI\xc0\xafK\xc1F'

b'\xc4\xc2\x1d\x1cj\xddAd'

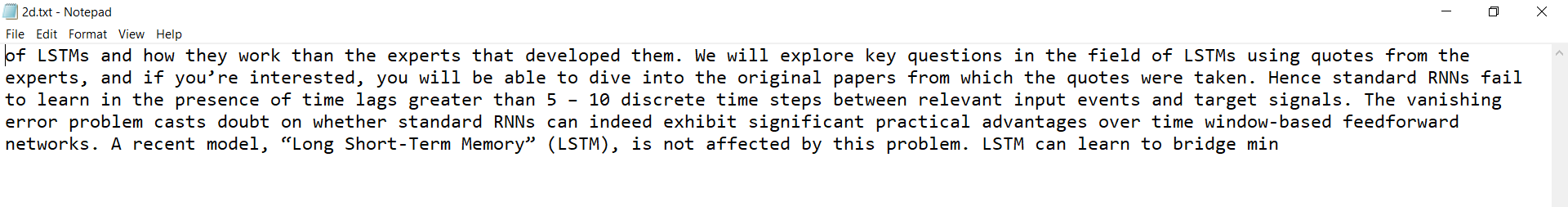
**5.4.2 Decryption of Data**

The extracted keys and the encrypted files unzipped from final.zip which is placed folder temp is used to extract the original data.

**5.4.2.1 Blowfish Decryption**

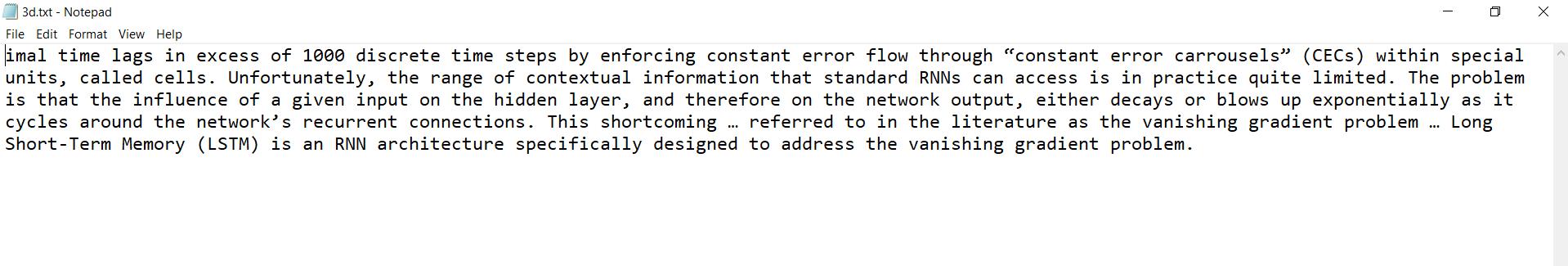
The keys and iv vector used for encryption is used here since Blowfish is a Symmetric key cryptography technology same keys are used for both encryption and decryption. The decrypted plain file obtained from Cipher text is ****

**5.4.2.2 AES Decryption**

The keys and iv vector used for encryption is used here since Advanced Encryption standard is a Symmetric key cryptography technology same keys are used for both encryption and decryption. The decrypted plain file obtained from Cipher text is 

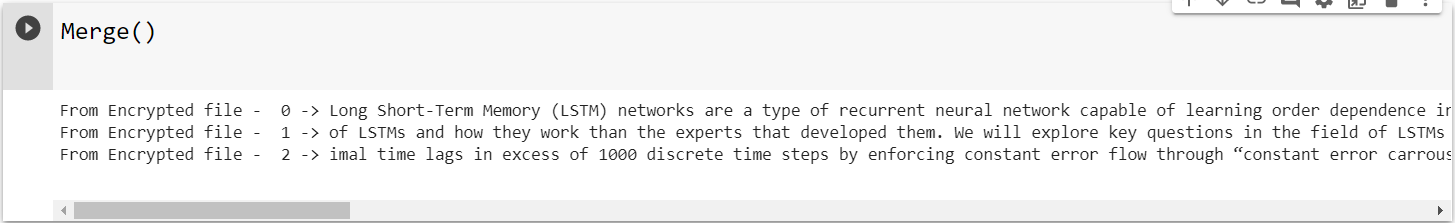
**5.4.2.3 TripleDES Decryption**

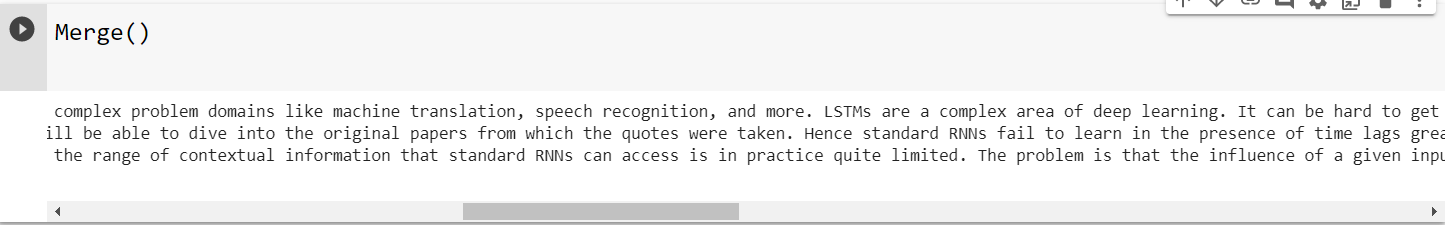
The keys and iv vector used for encryption is used here since Triple Data Encryption standard is a Symmetric key cryptography technology same keys are used for both encryption and decryption. The decrypted plain file obtained from Cipher text 3e.txt is

****

**5.4.3 Output Data**

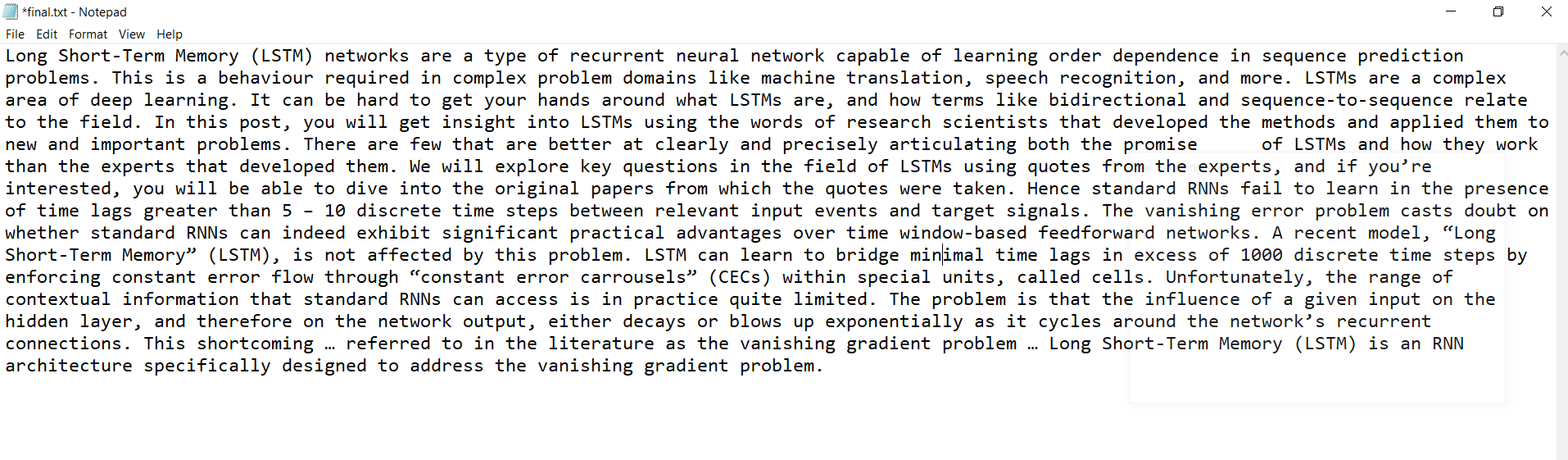
Data decrypted from three cryptography algorithms such as Blowfish, Advanced Encryption standard and Triple Data encryption standard are decrypted into three parts separately since we have applied hybrid cryptography each part is encrypted by separate algorithms now we have to combine those decrypted files so that we can obtain the original text back





**Original Data :**

**Merged** file is stored into a single text file called final.txt

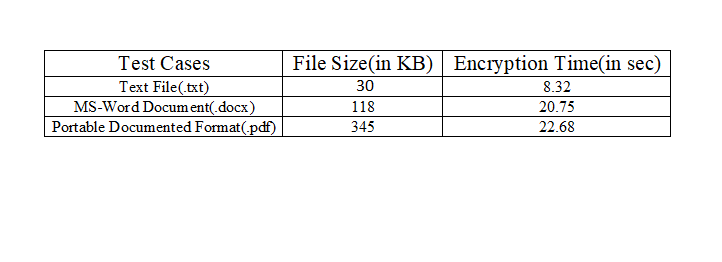
****

**Chapter 6**

**TEST CASES AND PERFORMANCE METRICS**

**6.1 TEST CASES**

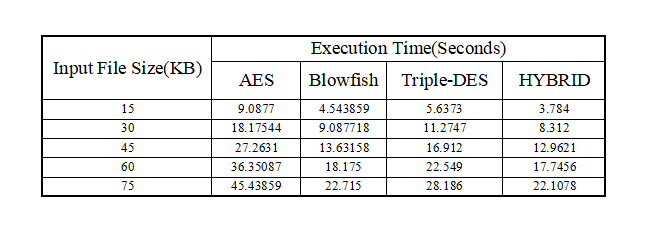
Files of different sizes and types were encrypted using the proposed system with the same password, for every file its encryption time is used to analyze the speed of the system.



**Table.No.6.1 Test cases**

**6.2 PERFORMANCE METRICS**

The metrics that are selected for the evaluation are encryption time, decryption time, throughput of encryption, throughput of decryption, diffusion analysis, CPU process time, and CPU clock cycles, power consumption and memory utilization. Here Encryption time is used to measure with existing system and hybrid system.



**Table.No.6.2 Performance Metrics**

**Chapter-7**

**7.1 Conclusion**

The proposed cryptosystem uses a combination of symmetric cryptography to secure data. The system also introduces a sub-process to encrypt the keysused forencryption before embedding them in an image. The combination of BlowfishAES-TripleDES has significantly improved the security and also ensured that the drawbacks

of the standalone systems are addressed. The system also helps in improving security

without the use of keys of larger lengths. We have also seen from the test results that thesystem is less susceptible to brute force attacks as the decryption time is significantlyhigh. The manyfold expansion of plaintext into ciphertext also helps in ensuring a highlevel of security. While the system successfully does its intended work, it still requiredminor improvements for larger adoption.

**7.2 Future Enhancement**

The proposed system is very secure and robust. It has proven to encrypt data and ensure key security. While it is efficient and secure, it was also seen that the encrypted files are generally 2-3 times the size of the original file, hence the encrypted file takes up a significant amount of space to store. This drawback can be addressed by studying it further and making changes to the proposed system. Another improvement that can be made is by making the encryption and decryption time lesser. Further research on the proposed system can also be done by analyzing different order of combinations of the three algorithms used. A slightly different combination can also be studied by replacing one of the algorithms for improved performance. System is now designed only for text based files future work is to make the system capable to support audio, video, many other file format.

**REFERENCES**

**1.Secure File Storage Using Hybrid Cryptography**:

2021 6th International Conference on Communication and Electronics Systems (ICCES)

**Authors:** Putta Bharathi, Gayathri Annam, Jaya Bindu, Kandi Vamsi, Krishna Duggana and Anjali.T

**2.An Efficient Algorithm for Confidentiality, Integrity and Authentication Using Hybrid Cryptography and Steganography 2019 :**

International Conference on Electrical, Computer and Communication Engineering ( ECCE), 7­9 February, 2019

**Authors:** Chitra Biswas, Udayan Das Gupta and Md. Mokammel Haque

**3.Hybrid Implementation of Twofish, AES, ElGamal and RSA Cryptosystems:**

2020 IEEE East­West Design &amp; Test Symposium (EWDTS)

**Authors:** Elza Jintcharadze Maksim Iavich

**4. Exploring LSB Steganography Possibilities in RGB Images:**

2021 12th International Conference on Computing Communication and Networking Technologies

**Authors:** Rutvik Dumre and Aashka Dave

**5.Combined Cryptography and Steganography for Enhanced Security in Suboptimal Images:** 2020 International Conference on Artificial Intelligence and Signal Processing (AISP)

**Authors**:S. Joseph Gladwin and Pasumarthi Lakshmi Gowthami

**6.Enhancing Data Security by using Hybrid Cryptographic Algorithm**

Authors, Publication & Year: Jigar Chauhan, Neekhil Dedhia, BhagyashriKulkarni ‐ IJESIT, 2013

**7.Performance Comparison Between AES 256‐Blowfish and Blowfish‐AES 256 Combinations**

**Authors:**Publication & Year:Muhammad Abdul Muin, Muhammad Abdul Muin, Arief Setyanto, Sudarmawan, Kartika Imam Santoso ‐ ICITACEE,2018

**8 .An Analytical Study for Some Drawbacks and Weakness Points of the AES Cipher (Rijndael Algorithm)** The 1st International Conference on Information Technology (ICoIT'17) At Lebanese French University Erbil, Kurdistan Region – Iraq 10th April, 2017 Authors: Omar A. Dawood, Othman I Hammadi