A MINOR PROJECT REPORT ON

**“MESSAGE ENCRYPTION & DECRYPTION”**

**Submitted**

*In the partial fulfilment of the requirements for*

*the award of the degree of*

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE & ENGINEERING**

By

**S.Deepak (171FA04428)**

**I.Rahul Ashish (171FA04462)**

**S.Pranathi (171FA04488)**

Under the guidance of

**Mr.D.Veeraiah**, Assistant.professor

****

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY AND RESEARCH UNIVERSITY**

(**Accredited by NAAC “A” grade**)

**Vadlamudi, Guntur.**

****

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEEING**

**CERTIFICATE**

This is to certify that the report entitled **“MESSAGE AND FILE ENCRYPTION & DECRYPTION”** is submitted by **S.Deepak (171FA04428), I.Rahul Ashish (171FA04462), S.Pranathi(171FA04488),** in the partial fulfilment of course work of Operating Systems as a minor project, carried out in the department of CSE , VFSTR deemed to be university.

Mr.Veeraiah, Mr.Venkatesulu,

Assistant Professor External Examiner H.O.D,CSE

**DECLARATION**

We hereby declare that the Minor project entitled “MESSAGE AND FILE ENCRYPTION &DECRYPTION” submitted for the **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**. This dissertation is our original work and the project has not formed the basis for the award of any degree, associate-ship and fellowship or any other similar titles and no part of it has been published or sent for publication at the time of submission.

By

S.Deepak

I.Rahul ashish

S.Pranathi

**INDEX**

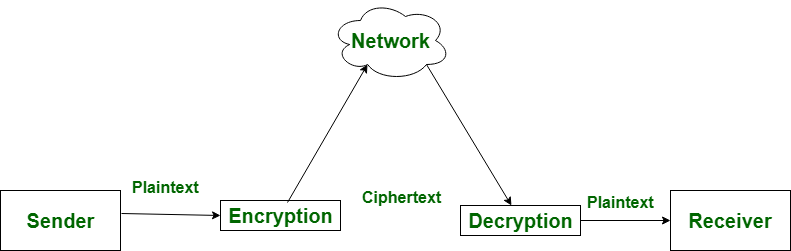
1. **Abstract**
2. **Introduction**
3. **Requirements**
4. **Methodology**
5. **Algorithm**
6. **Implementation**
7. **Output**
8. **Conclusion**
9. **Advantages**
10. **Disadvantages**
11. **References**

**ABSTRACT**

Encryption is process of turning a plaintext to jargon or the method of changing confidential file to jargon in order prevent unauthorized persons to gain access to confidential message. Message is the transfer of information from the sender to the receiver through a particular medium. Encryption is the most effective process for achieving data security. The process of Encryption hides the contents of a message in a way that the original information is recovered only through a decryption process.

**INTRODUCTION**

Encryption is the process of converting normal message (plain text) into meaningless message (Cipher text). Decryption is the process of converting meaningless message (Cipher text)into its original form (Plain text).

****

**Abstract:**Tetfund Sponsored Kwara State Polytechnic Journal of Research and Develoment Studies Vol. 5.

No. 1 June 2017

**REQUIREMENTS**

**Software requirements**

* Editor Spyder
* Compiler python 3.7
* Os windows

**Hardware requirements**

* Processor i3or above
* Ram 2gb or above
* Rom 256gb or above

**METHODOLOGY**

* This program is having two option those are

1. Message encryption or decryption.
2. File encryption and decryption.

* We should chose any of them,after that we should give the input for encryption or decryption.
* In this program the we used some modules by which the process of encryption and decryption is done easily.
* By importing those modules we can use some inbuild functions like encrypt(),decrypt() etc.,
* For encrypting or decrypting we should give some password which acts as a key for convertion.
* If the password given at encryption and decryption are not same then the message can not be converted.

**ALGORITHM**

Step1:import required modules.

Step2:read a variable to ensure that it is file encryption or message encryption

Step3:define encrypt function and write code for encryption

Step4:define decrypt function and write code for decryption

Step5:write a loop to repeat the process.

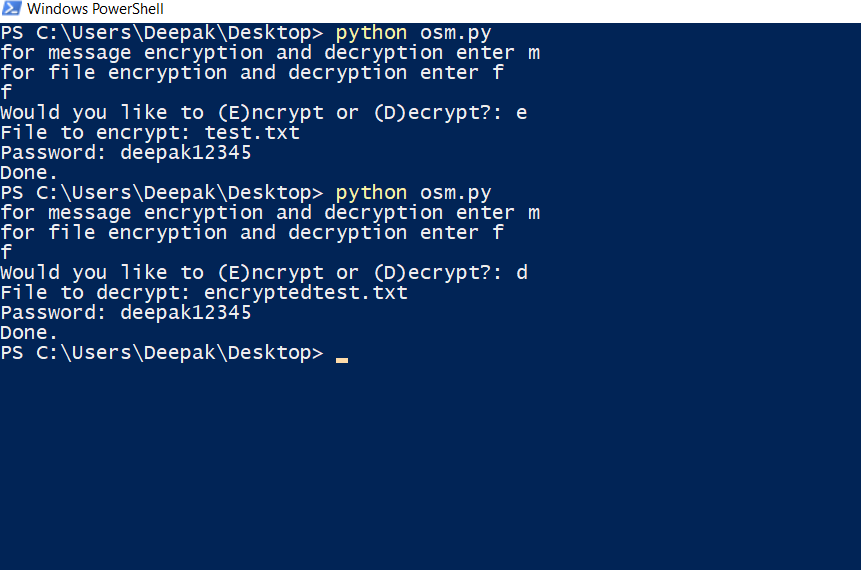
Step6:if the process is completed exit.

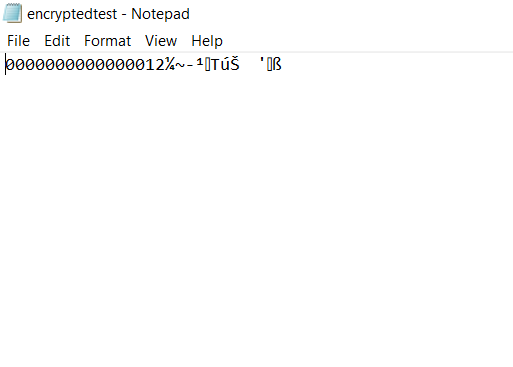
**IMPLEMENTATION**

import os  
from Crypto.Cipher import AES  
from Crypto.Hash import SHA256  
from Crypto import Random  
import base64  
import hashlib  
from Crypto.Cipher import AES  
print("for message encryption and decryption enter m\nfor file encryption and decryption enter f")  
h=input()  
if(h=='m'or h=='M'):  
    BLOCK\_SIZE = 16  
    pad = lambda s: s + (BLOCK\_SIZE - len(s) % BLOCK\_SIZE) \* chr(BLOCK\_SIZE - len(s) % BLOCK\_SIZE)  
    unpad = lambda s: s[:-ord(s[len(s) - 1:])]  
    i=input("enter message:")  
    password = input("Enter encryption password: ")  
     
       
    def encrypt(raw, password):  
  
        private\_key = hashlib.sha256(password.encode("utf-8")).digest()  
        raw = pad(raw)  
        iv = Random.new().read(AES.block\_size)  
        cipher = AES.new(private\_key, AES.MODE\_CBC, iv)  
        return base64.b64encode(iv + encrypt(raw))  
       
       
    def decrypt(enc, password):  
  
cipher.        private\_key = hashlib.sha256(password.encode("utf-8")).digest()  
        enc = base64.b64decode(enc)  
        iv = enc[:16]  
        cipher = AES.new(private\_key, AES.MODE\_CBC, iv)  
        return unpad(cipher.decrypt(enc[16:]))  
       
    while(True):  
        h=input("enter e for encryption and d for decryption: ")  
        if h=='e':  
            encrypted = encrypt(i, password)  
            print(encrypted)  
           
        if h=='d':  
            decrypted = decrypt(encrypted, password)  
            print(bytes.decode(decrypted))  
            break  
if(h=='f'or h=='F'):          
    def encrypt(key, filename):  
        chunksize =64 \* 1024  
        outputFile = "encrypted"+filename  
        filesize = str(os.path.getsize(filename)).zfill(16)  
        IV = Random.new().read(16)  
         
        encryptor = AES.new(key, AES.MODE\_CBC, IV)  
         
        with open(filename, 'rb') as infile:  
            with open(outputFile, 'wb') as outfile:  
                outfile.write(filesize.encode('utf-8'))  
                outfile.write(IV)  
         
                while True:  
                    chunk = infile.read(chunksize)  
                 
                    if len(chunk) == 0:  
                        break  
                    elif len(chunk) % 16 != 0:  
                        chunk +=b' ' \* (16 - (len(chunk) % 16))  
             
                outfile.write(encryptor.encrypt(chunk))  
    def decrypt(key, filename):  
        chunksize =64 \* 1024  
        outputFile = filename[11:]  
         
        with open(filename, 'rb') as infile:  
            filesize = int(infile.read(16))  
            IV = infile.read(16)  
         
            decryptor = AES.new(key, AES.MODE\_CBC, IV)  
             
            with open(outputFile, 'wb') as outfile:  
                while True:  
                    chunk = infile.read(chunksize)  
             
                    if len(chunk) == 0:  
                        break  
             
                    outfile.write(decryptor.decrypt(chunk))  
                     
         
    def getKey(password):  
        hasher = SHA256.new(password.encode('utf-8'))  
        return hasher.digest()  
     
    def Main():  
        choice = input("Would you like to (E)ncrypt or (D)ecrypt?: ")  
        if choice == 'E' or choice == 'e':  
            filename = input("File to encrypt: ")  
            password = input("Password: ")  
            encrypt(getKey(password), filename)  
            print("Done.")  
        elif choice=='d' or choice=='D':  
            filename = input("File to decrypt: ")  
            password = input("Password: ")  
            decrypt(getKey(password), filename)  
            print("Done.")  
        else:  
            print("No Option selected, closing…")  
     
    if \_\_name\_\_ == '\_\_main\_\_':  
        Main()

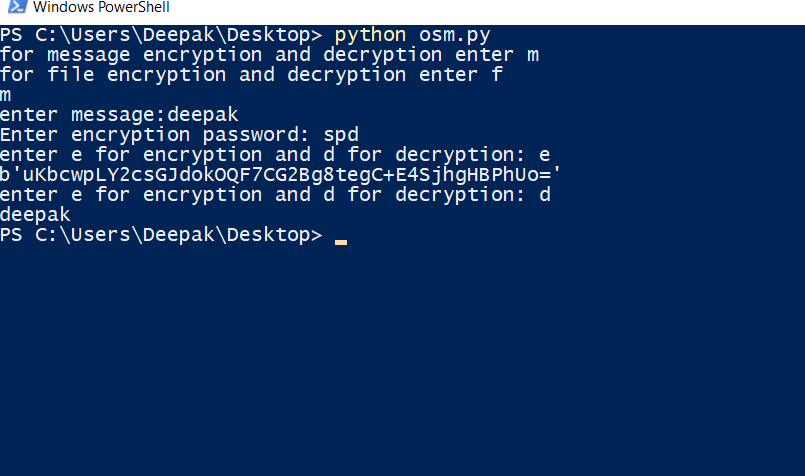
**OUTPUT**

**For file encryption and decryption:-**

****



**For message encryption and decryption:-**



**CONCLUSION**

* By this program we provide security to message by that unauthorized persons can’t view the original message

**Advantages**

* Provides security
* File encryption
* Message encryption
* Password manager
* Easy implementation

**Disadvantage**

* Decryption of file is not possible with this code
* Difficult to understand.
* Modules used are not provided by IDE

**REFERENCE**

<https://www.tutorialspoint.com/cryptography_with_python/cryptography_with_python_quick_guide.htm>

<https://www.novixys.com/blog/using-aes-encryption-decryption-python-pycrypto/>

<https://nitratine.net/blog/post/encryption-and-decryption-in-python/>