



# **Securing File Storage on Cloud using Hybrid Cryptography**

**Mohamed Ali Mohamed**

**Mostafa Mahmoud Mostafa**

**Ahmed Mostafa Morad**

**Mohamed Hani Abou El-Ela**

**Supervisor:** Dr. Sahar Abd El-Rahman

**Department of Computer Systems Engineering**

**Faculty of Engineering - Shoubra**

**Benha University**

**2021/2022**

# Contents

Abstract & Keywords . . . . .	iv
<b>1 Introduction</b>	<b>1</b>
1.1 Report Objective . . . . .	1
1.2 Background . . . . .	1
1.3 Literature Review . . . . .	2
1.3.1 Cloud Computing [9] . . . . .	2
1.3.2 Cryptography . . . . .	2
1.3.3 Steganography . . . . .	3
1.4 Project Scope . . . . .	4
1.5 Project Objectives . . . . .	4
1.6 Proposed Solution . . . . .	4
<b>2 Design</b>	<b>5</b>
2.1 System Architecture . . . . .	5
2.1.1 Application Layer: . . . . .	5
2.1.2 Cloud Layer: . . . . .	5
2.2 Cryptography Algorithm . . . . .	6
2.2.1 Symmetric Encryption: . . . . .	6
2.2.2 Asymmetric Encryption: . . . . .	6
2.3 Core System Operations . . . . .	7
2.3.1 Encryption Process: . . . . .	7
2.3.2 Uploading and storing the encrypted file on AWS Cloud Process: [1] . . . . .	7
2.3.3 Image Steganography Process: . . . . .	8
2.3.4 Sending Decryption keys to the user through E-mail: . . . . .	8
2.3.5 Decryption process: . . . . .	8
2.3.6 Listing stored files in the S3 bucket: . . . . .	9
<b>3 Implementation</b>	<b>10</b>
3.1 Encryption Process . . . . .	10
3.2 Uploading and storing the encrypted file on AWS Cloud Process . . . . .	11
3.3 Steganography Process . . . . .	11
3.4 Sending Decryption keys to the user through E-mail . . . . .	11
3.5 Decryption process . . . . .	12
3.6 Listing stored files in the S3 bucket . . . . .	12

<b>4</b>	<b>Testing</b>	<b>13</b>
4.1	How a user sets up an AWS account . . . . .	13
4.2	How a user runs our program . . . . .	13
4.3	How a user can interact with our program . . . . .	17
4.4	How a user can handle any errors due to entering a wrong argument . . . . .	20
	<b>Bibliography</b>	<b>21</b>

---

# Abstract

Cloud computing is used in many areas like industry and military for storing huge amounts of data. To keep the data in a more secure form, we encrypt the file before uploading to the cloud server and do not store the decryption key on the cloud. Using a single algorithm in the encryption process is not sufficient for high level security of data in cloud computing. Therefore, we introduce a security solution using hybrid cryptography algorithms and steganography to achieve data integrity, security, confidentiality. We use symmetric cryptography algorithm for encrypting the file content and the public key cryptosystem (PKC) for encrypting the symmetric key itself. Finally, the decryption key must be saved securely in a remote place of the files themselves, so image steganography is used to hide the key in a cover image and then send the carrier (stego) image that contains the decryption key to the receiver through email. For the decryption process the reverse process of encryption is applied.

**Keywords**— Security, Cryptography, Steganography, Hybrid Cryptography, Cloud

# Chapter 1

## Introduction

### 1.1 Report Objective

In this report, we introduce a solution to achieve the secrecy of the data during uploading to, downloading from, and storing on the cloud then saving the decryption key secure through image steganography. This solution combines both symmetric and asymmetric cryptography and steganography to achieve data integrity, security, confidentiality.

### 1.2 Background

Nowadays most people are very interested in sending and receiving data through internet and mobile data storage devices. Although this data usually contains personal information, a great number of them do not encrypt their data therefore the chances of data lose, or hacking are very high. The importance of Information security has become in continuous growth in all aspects of life. As long as the technology continues to control various operations in our day-to-day life, this increases the need for information security. When the medium of information transmission is susceptible to interception, we use cryptography to provide a layer of security through translating messages into a form that cannot be understood by an unauthorized user or third party. An efficient solution for storing data in a secure form, is cloud storage where data is stored on Internet-connected servers instead of local hard drives. These servers are managed by data centers to keep the data safe and secure to access. Cloud-based internet security is efficient solution for storing data. While data moves between your local device and the cloud provider, encryption in transit is used to protect your data if the communication are intercepted by a malicious actor. This protection is achieved through encrypting the data before transmission using symmetric cryptosystem and using public key cryptosystem (PKC) for key exchange processes between the user and the cloud.

## 1.3 Literature Review

### 1.3.1 Cloud Computing [9]

The "cloud" refers to servers which are accessed over the Internet, and the software programs and databases that run on those servers. Cloud servers are placed in data centers all over the world. Through cloud, customers and companies do not have to manipulate physical servers themselves or run software applications on their own machines. Users can access the same files and applications from almost any device through the cloud because the computing and storage occur on servers in a data center, instead of local devices of the users. Switching to cloud computing gets rid of some IT costs and overhead for businesses. For example, businesses no longer need to update and maintain their own servers because the cloud vendor, with whom they are subscribed, will do that. Small businesses that may not have been able to afford their own internal infrastructure can now deploy their infrastructure on the cloud with low cost. Another benefit of the cloud for companies is that it makes it easier for them to operate internationally, because employees and customers can access the same files and applications from any location.

### 1.3.2 Cryptography

Cryptography is the practice and study of techniques for secure communication in the presence of adversarial behavior. Cryptography is about constructing and analyzing protocols that prevent third parties or the public from reading private messages. Various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation are central to modern cryptography [4]. Modern cryptography involves the study of mathematical techniques for securing digital information, systems, and distributed computations against adversarial attacks [12].

- **Symmetric-key cryptography:** refers to the methods in which both the sender and receiver use the same key for encryption and decryption. Symmetric key ciphers are implemented as either block ciphers or stream ciphers [4]. In a block cipher, the plaintext is divided into fixed-sized chunks called blocks. A block is specified to be a bit string (i.e., a string of 0's and 1's) of some fixed length (e.g., 64 or 128 bits). A block cipher will encrypt (or decrypt) one block at a time. The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are examples of block cipher algorithms. In contrast, a stream cipher first uses the key to construct a keystream, which is a bit string that has exactly the same length as the plaintext (the plaintext is a bitstring of arbitrary length). The encryption operation constructs the ciphertext as the exclusive-or of the plaintext and the key stream. Decryption is accomplished by computing the exclusive-or of the ciphertext and the key stream [14]. There are many symmetric-key algorithms, such as RC4 which is an example of stream cipher algorithms.
- **Asymmetric-key cryptography:** which is called public-key cryptography refers to the methods in which the sender and receiver use two different but mathematically related keys for encryption and decryption. These two keys are called public key, which is used for encryption, and private key, which is used for decryption. A Public key is known for everyone, whereas a private key is known only for the recipient of

the encrypted message. Asymmetric-key cryptography is a modern cryptography technique that is more secure and robust than symmetric cryptography, but it is slower than symmetric cryptography requires high consumption of resources and used to encrypt small amount of data [8]. There are many asymmetric-key algorithms such as Diffie–Hellman, RSA, elliptic curve and ElGamal encryption.

- **Hybrid cryptography:** is to combine symmetric and asymmetric cryptography together to benefit from the strengths of each cryptosystem. Hybrid cryptography (almost) achieves the efficiency because it uses the slow asymmetric cryptosystem to encrypt the short secret key and the symmetric cryptosystem to encrypt the longer plaintext [14].

### 1.3.3 Steganography

Steganography is the practice of concealing a message within another message or a physical object [7]. The use of steganography can be combined with encryption as an extra step for hiding or protecting data. The advantage of steganography over cryptography alone is that the intended secret message does not attract attention to itself as an object of scrutiny. The content to be concealed through steganography is often encrypted before being incorporated into the innocuous-seeming cover text file or data stream [13]. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program, or protocol. The ideal form for steganographic transmission is media files because they have a large size. For example, a sender might start with an innocuous image file and adjust the color of every hundredth pixel to correspond to a letter in the alphabet. The change is so subtle that someone who is not specifically looking for it is unlikely to notice the change [7].

Steganography comes in five types, text steganography, audio steganography, video steganography, image steganography and network steganography [11]. Hiding data within an image is called image steganography. It depends on two images, the first is called the cover image in which we hide the data, and the other is called the stego image, it is the image that is obtained after steganography. To embed a message in an image you need to alter the values of some pixels which are chosen by an encryption algorithm. To extract the message, the recipient of the stego image must be aware of the same algorithm to know which pixel he/she must select to extract the message [10].

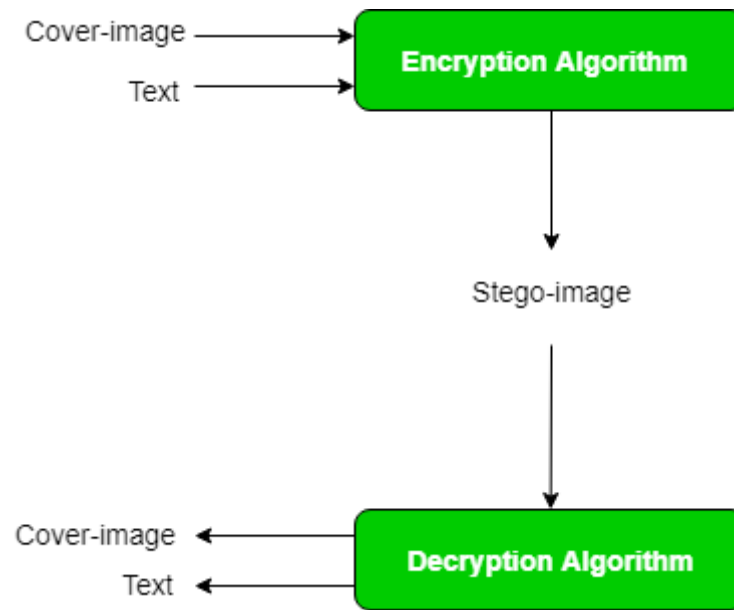


Figure 1.1: Steganography [10].

## 1.4 Project Scope

We will implement a hybrid cryptography solution using symmetric and asymmetric cryptography algorithms that allow the users to upload and download their files from cloud in an encrypted form for backing up purposes and storing the data securely. Finally, we will use steganography to send the decryption key to the user in a hidden form.

## 1.5 Project Objectives

- Using cloud storage for storing data in a secure form
- Achieve the secrecy of data during sending to the cloud using hybrid cryptography algorithms
- Send the decryption key to the user in in a hidden form

## 1.6 Proposed Solution

We propose a solution to achieve the secrecy of the data during uploading to and downloading from the cloud then saving the decryption key secure through steganography.



# Chapter 2

## Design

### 2.1 System Architecture

Our system consists of two layers:

1. Application Layer.
2. Cloud Layer.

#### 2.1.1 Application Layer:

In this layer, the operation of encryption and decryption are performed on the local computer of the authorized users.

#### 2.1.2 Cloud Layer:

In this layer, all encrypted files are stored to keep them in a secure environment that can be accessed through the authorized users.

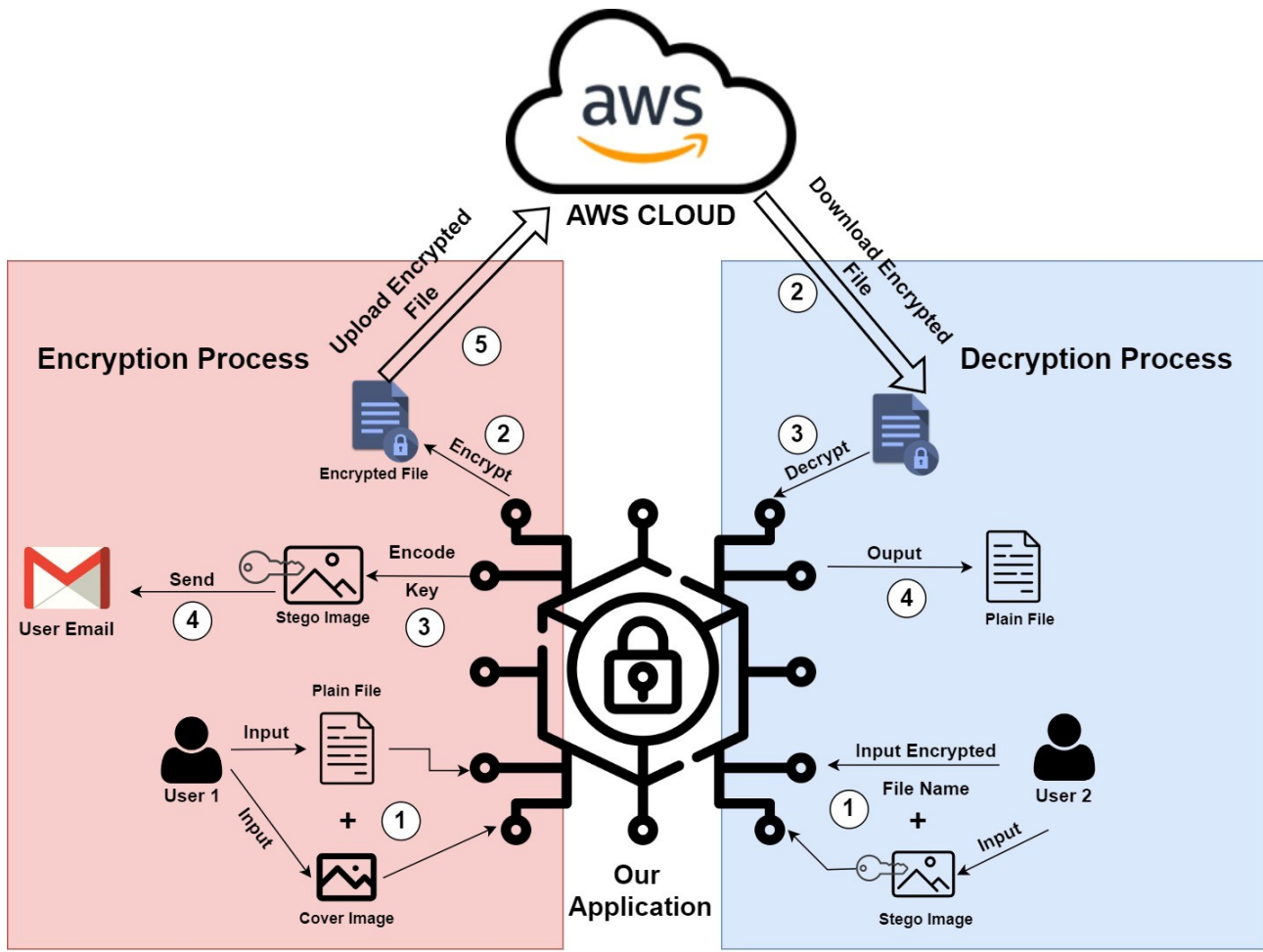


Figure 2.1: System Architecture.

## 2.2 Cryptography Algorithm

Hybrid cryptography is used in our system because using a single algorithm in the encryption and decryption processes is not sufficient for high level security of data, so symmetric and asymmetric algorithms are used to achieve data integrity, security, confidentiality.

### 2.2.1 Symmetric Encryption:

Advanced Encryption Standard (AES) algorithm with a key length of 256 bits and in CBC mode is used for encrypting and decrypting the content of the file to be encrypted.

### 2.2.2 Asymmetric Encryption:

RSA (Rivest–Shamir–Adleman) algorithm with a key length 1024 bits is used in the process of encrypting and decrypting the symmetric key.

## 2.3 Core System Operations

### 2.3.1 Encryption Process:

- **Input:** A user enters a file of any extension (text, image, video, audio, etc.) to be encrypted.
- **Output:** An encrypted file.

This process consists of a set of sub-processes as follows:

1. Generating public and private key of the RSA algorithm.
2. Generating the symmetric key of the AES algorithm.
3. Encrypting the file content with the symmetric key.
4. Encrypting the symmetric key with the public key.
5. collecting the private key and the encrypted symmetric key in a dictionary which is called “info” to use them later.

### 2.3.2 Uploading and storing the encrypted file on AWS Cloud Process: [1]

To increase the security and availability, the encrypted files are stored on Amazon World Services cloud. Depending on Amazon Simple Storage Service (Amazon S3) which is an object storage service we guarantee a secure and reliable environment for keeping the encrypted files.

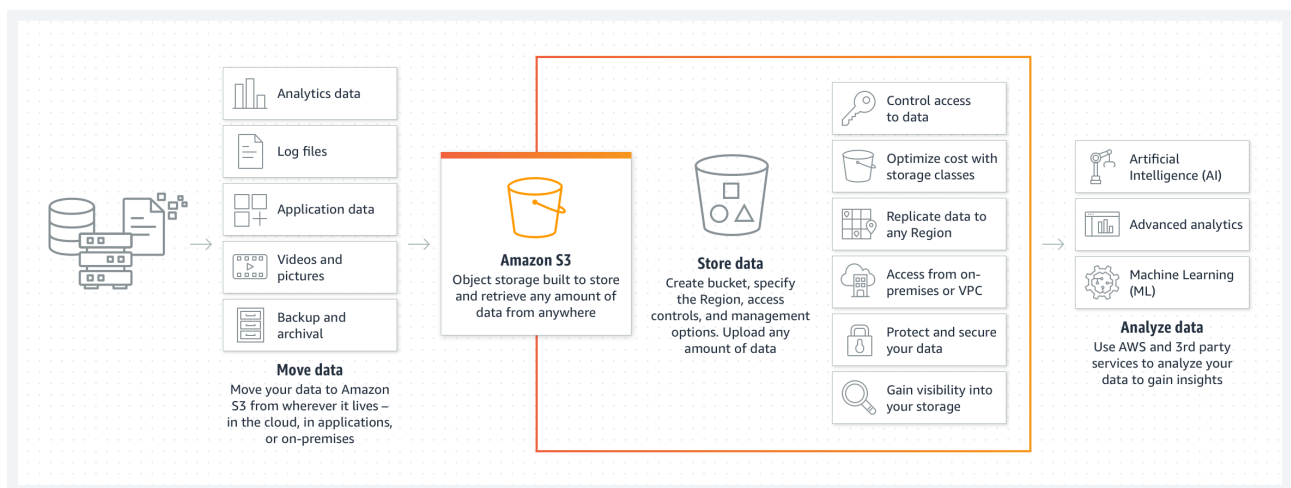


Figure 2.2: Amazon Web Service - S3 [1].

An S3 bucket is created, called “securestore1” and used to store the encrypted files. After each encryption process the application accesses this bucket to which the encrypted file is uploaded and sorted.

### 2.3.3 Image Steganography Process:

- **Input:** A user enters a cover image to be used in the encoding operations.
- **Output:** A stego image that contains the required information for decryption process.

Image steganography are used to encode and hide the private key of the RSA algorithm and the encrypted symmetric key in the cover image given by the user. To implement this process, first we convert the private key and the encrypted symmetric key to a stream of bits, then for every pixel of the cover image that consists of red, green, blue channels and every channel of them is stored in a byte, the least significant bit of that byte is replaced with a bit from the bits of the info dictionary. Therefore, the cover image should contain a sufficient number of pixels to achieve the previous operation successfully. To know the number of bytes that the cover image can encode we depend on the following equation:

$$max-bytes-number = (image-pixel-numbers \times 3) / 8 \quad (2.3.1)$$

After this process is finished the output image is called the Stego image, and it is send to the authorized user to provide the required keys for decryption process.

### 2.3.4 Sending Decryption keys to the user through E-mail:

After producing the stego image that contains the required information for decryption process, this stego image is sent to the authorized user who has the rights to accesses the file and decrypts it to view its content.

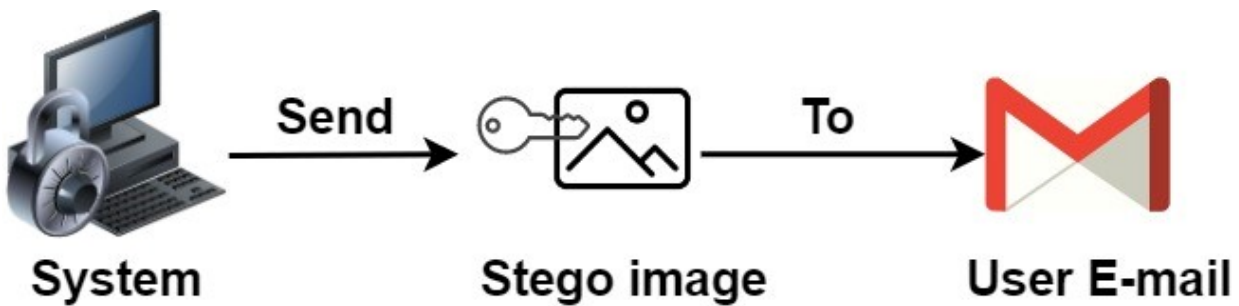


Figure 2.3: Steganography.

### 2.3.5 Decryption process:

- **Input:** The user enters the encrypted file name.
- **Output:** The original file.

This process consists of a set of sub-processes as follows:

1. Extracting the private key and the encrypted symmetric key from the stego image.
2. Decrypting the encrypted symmetric key through the private key.

3. Downloading the encrypted file from the S3 bucket.
4. Decrypting this file through the symmetric key.
5. Return the original file to the user.

### **2.3.6 Listing stored files in the S3 bucket:**

An additional service besides the cryptographic services is that the system enables the user to list all encrypted files on the S3 bucket such that the user can know which files are encrypted and stored. This service is very useful specially before the decryption operation because the user during the decryption operation needs only to enter the name of the encrypted file to the system and the system will download the file from the cloud and decrypt it.

# Chapter 3

## Implementation

As we discussed the different processes in the system through the design chapter, we will discuss how every process is implemented.

- **Used programming languages:** Python
- **Main Modules:**

Table 3.1: Used Python Modules.

Module Name	Usage Purpose
boto3	The Amazon Web Services (AWS) Software Development Kit (SDK) for Python, that is used to upload and download files from S3 buckets
number	Used to generate prime numbers for RSA algorithm.
argparse	Allows user input values to be parsed and utilized. through incorporating the parsing of command line arguments. Instead of having to manually set variables inside of the code.
secrets	Used for generating symmetric key.
pyAesCrypt	Uses AES256-CBC to encrypt/decrypt files.
stegano	Used to apply Least Significant bit steganography.
smtplib	Used for sending emails using the Simple Mail Transfer Protocol (SMTP).

### 3.1 Encryption Process

1. To generate the public key (n,e) we use generate\_key() function that generates two large primes p and q then it calculates the value of n from the product of q and p and the value of  $\phi$  from the product of p-1 and q-1 values. The exponent e is a random number in the range  $[1, \phi]$  and usually its value is 65537.
2. After we calculate the value of exponent e, the value of d needs to be found to produce the private key (d, e) so we use the multiplicative\_inverse() function that depends on the Extended Euclidean Algorithm to calculate the value of d depending on the equation:  $d \cdot e \equiv 1 \pmod{\phi(n)}$
3. Symmetric key generation is done by using the function token\_hex() from the secrets module. This function returns return a random text string, in hexadecimal of length of length 64 which is equivalent to 256 bits.

4. The file is encrypted by this symmetric key through `encryptFile()` function from the `pyAesCrypt` module. This function use AES-CBC encryption and takes file name , encrypted file name and the symmetric key as arguments and returns the encrypted file.
5. The next step is to encrypt the symmetric key itself by using the public key. This is done through `RSA_encrypt()` function which implements the equation:  $Y = X^e \text{ mode } n$  where  $X$  represents the symmetric key,  $(e,n)$  is the public key and  $Y$  is the encrypted symmetric key. `RSA_encrypt()` function depends on a `fast_expo()` function that implements the square-and-multiply algorithm that allows fast exponentiation, even with very long numbers.
6. At the end of the encryption process we collect the public, private and encrypted symmetric key in a dictionary called `info` to use them later in the process of steganography.

## 3.2 Uploading and storing the encrypted file on AWS Cloud Process

To store the encrypted file on S3 bucket, `bucket_upload_file()` is used. This function depends on `boto3` module that is the Amazon Web Services (AWS) Software Development Kit (SDK) for Python which is used to upload and download files from S3 buckets. The function takes the encrypted file name with `(.enc)` extension, bucket name and returns true if the file is stored in the bucket successfully and false if not. It uses the `boto3.client()` function to create a low level service client by name using the default session to connect the S3 bucket.

## 3.3 Steganography Process

Image Steganography is implemented to hide the content of `info` dictionary in the cover image entered by the user. This is done by the `lsb_hide()` function that depends on the `lsb` module from `stegano` package and takes the cover image name, the stego image name and the `info` dictionary as arguments to return the stego image which contains the public, private and the encrypted symmetric key hidden.

## 3.4 Sending Decryption keys to the user through E-mail

`send_email()` function is used to send the stego image that contains the necessary information for decryption process to the authorized user. it takes the email to which it sends the stego image and returns true if the email is sent successfully or false if the email is not send. This function depends on `email` package to create the message payload and header and `smtplib` module to create SMTP session for sending the mail.

### 3.5 Decryption process

1. The first step in this process is extracting the private key and the encrypted key from the stego image. This is achieved by using the `lsb_extract()` function that takes the stego image as an argument and returns a dictionary contains the private key and the encrypted key. This function depends on the `lsb` module specially on its `reveal ()` function that returns the hidden message in the image.
2. The second step is to decrypte the symmetric key by the private key that is extracted from the stego image. `RSA_decrypt()` function is used to implement that based on the equation:  $X = Y^d \bmod n$ , where  $X$  is the symmetric key and  $Y$  is the encrypted symmetric key. This function also depends on `fast_expo()` function that implements the square-and-multiply algorithm that allows fast exponentiation, even with very long numbers.
3. After gaining the symmetric key. the encrypted file needs to be download from the S3 bucket. First, a session with S3 is create through `boto3.client()` function then the `s3.download_file()` function is called to download the encrypted file.
4. Now we have the encrypted file and the symmetric key, so it is high time for the decryption process. from `pyAesCrypt` module we use `decryptFile()` function. It takes the encrypted file name , the new name for the original file and the symmetric encryption key.

### 3.6 Listing stored files in the S3 bucket

To enable the user for listing all files stored in the S3 buckets the `resource()` and `Bucket()` functions from the `boto3` module are used. The user enters the bucket name then through these functions the program lists all the stored file in this bucket.



# Chapter 4

## Testing

In this chapter the following topics are discussed:

- How a user sets up an AWS account.
- How a user runs the program.
- How a user can interact with the program.
- How a user can handle any errors due to entering a wrong argument.

### 4.1 How a user sets up an AWS account

For setting up AWS S3 for uploading and downloading files from the bucket you first need to setup your AWS account and create a bucket.

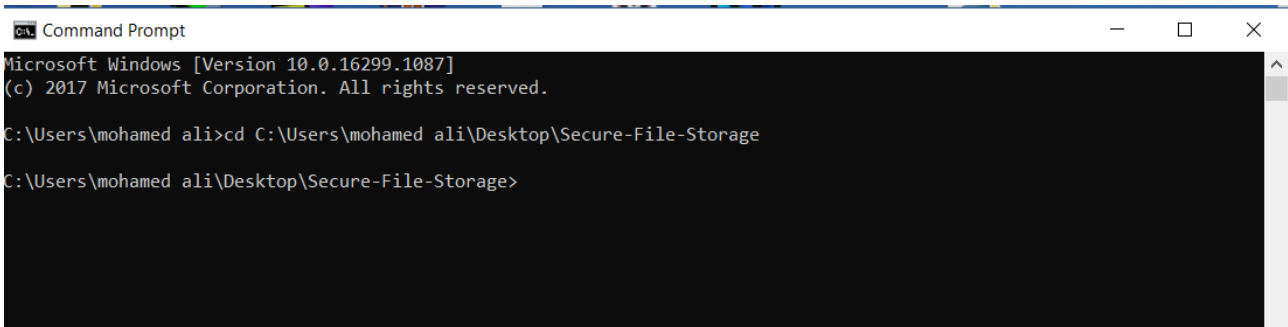
Before starting you need to create an AWS account if you don't have an account, by getting in [3] and create an account.

Then the user needs to set up an S3 bucket to be able to download and upload files from it, To do that you can follow [2] For creating a bucket and this one to set it up [6]

### 4.2 How a user runs our program

- A user needs to install python 3 on its local computer from this website [5] and chooses the suitable version for his operation system ( Window / Linux / Unix / macOS).
- After installing python, a user needs to install a set of packages listed in the requirements.txt file which contains the required packages to run our program through the command ***pip install -r requirements.txt***
- **main.py** is the basic file that a user needs to run it to operate the program is “main.py” file. This can be done with one of the following ways:

1. **Using Command prompt (CMD) or a terminal:** in this way a user needs to open the CMD or terminal in the same path of project directory



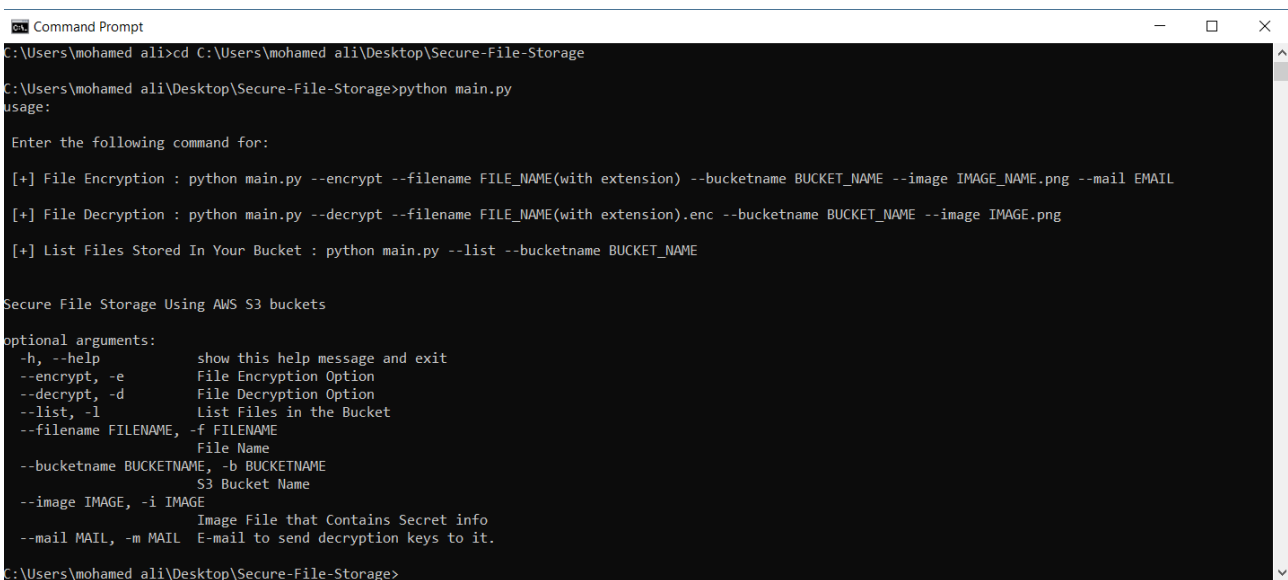
```

Microsoft Windows [Version 10.0.16299.1087]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\mohamed ali>cd C:\Users\mohamed ali\Desktop\Secure-File-Storage
C:\Users\mohamed ali\Desktop\Secure-File-Storage>

```

Then, a user use `>python main.py` command for Windows operating system or `>python3 main.py` command for Linux operation system to run the program.



```

C:\Users\mohamed ali>cd C:\Users\mohamed ali\Desktop\Secure-File-Storage
C:\Users\mohamed ali\Desktop\Secure-File-Storage>python main.py
usage:

Enter the following command for:

[+] File Encryption : python main.py --encrypt --filename FILE_NAME(with extension) --bucketname BUCKET_NAME --image IMAGE_NAME.png --mail EMAIL
[+] File Decryption : python main.py --decrypt --filename FILE_NAME(with extension).enc --bucketname BUCKET_NAME --image IMAGE.png
[+] List Files Stored In Your Bucket : python main.py --list --bucketname BUCKET_NAME

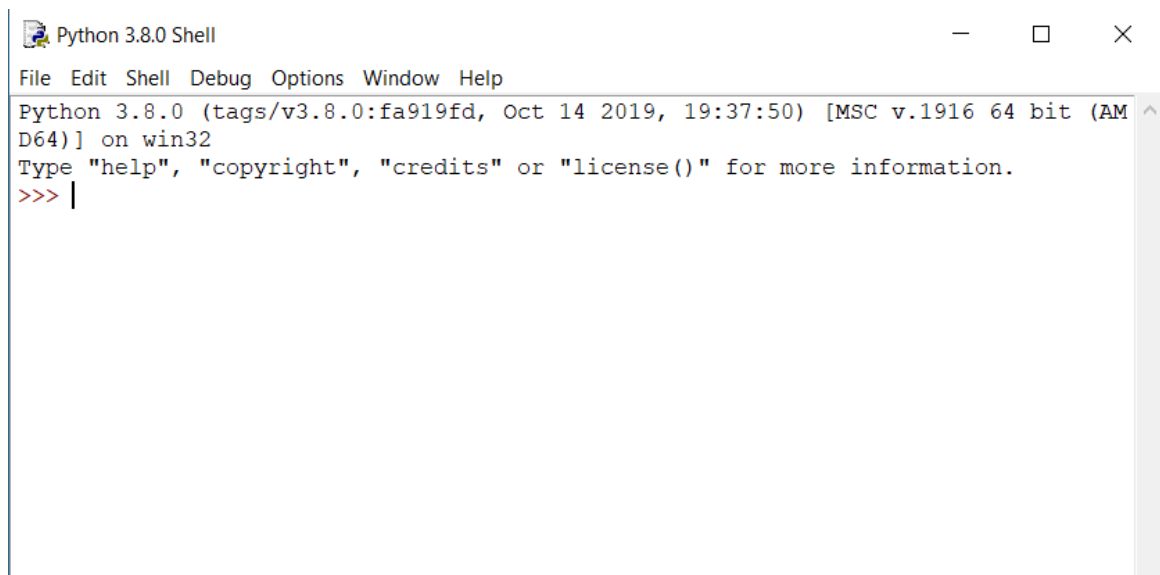
Secure File Storage Using AWS S3 buckets

optional arguments:
  -h, --help            show this help message and exit
  --encrypt, -e          File Encryption Option
  --decrypt, -d          File Decryption Option
  --list, -l            List Files in the Bucket
  --filename FILENAME, -f FILENAME
                        File Name
  --bucketname BUCKETNAME, -b BUCKETNAME
                        S3 Bucket Name
  --image IMAGE, -i IMAGE
                        Image File that Contains Secret info
  --mail MAIL, -m MAIL  E-mail to send decryption keys to it.

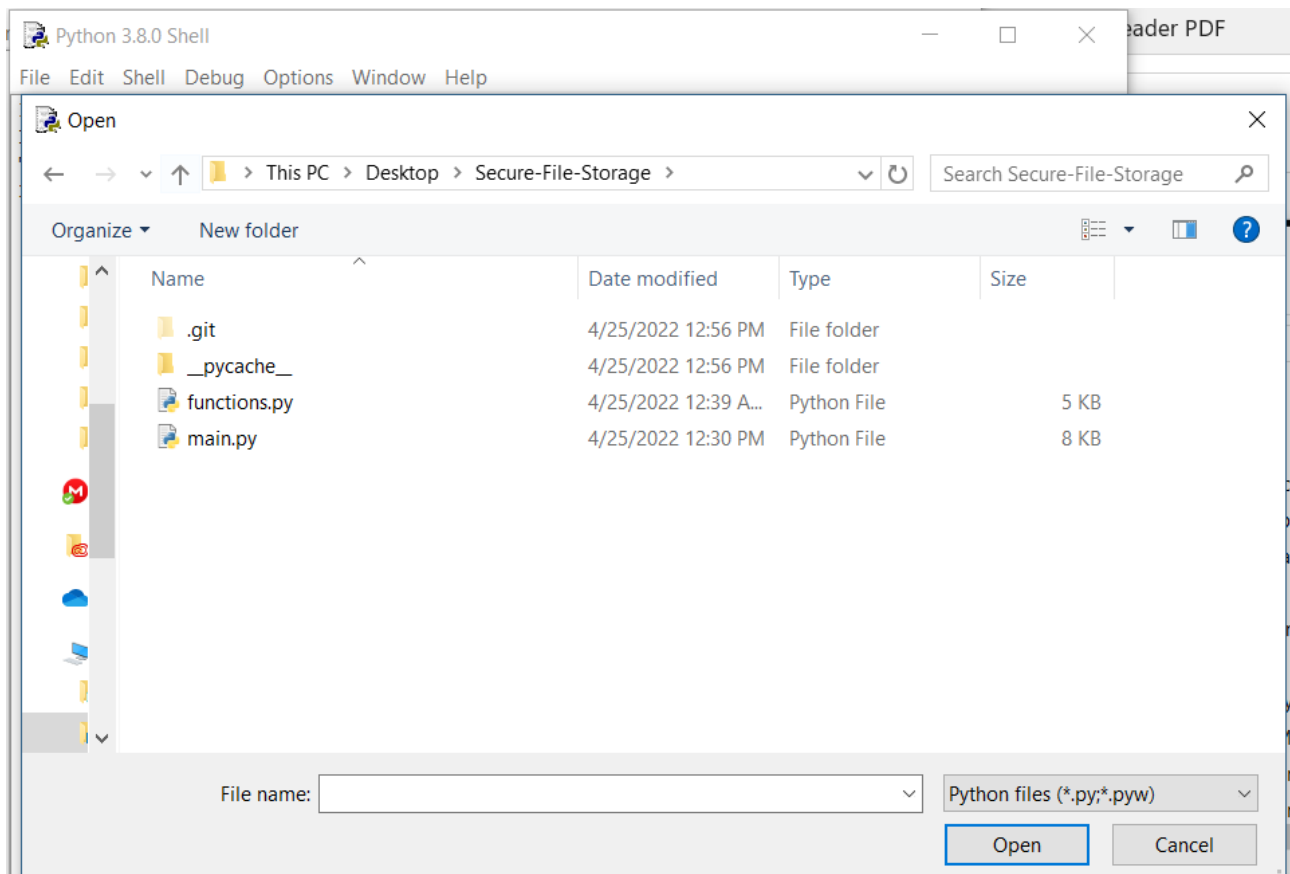
C:\Users\mohamed ali\Desktop\Secure-File-Storage>

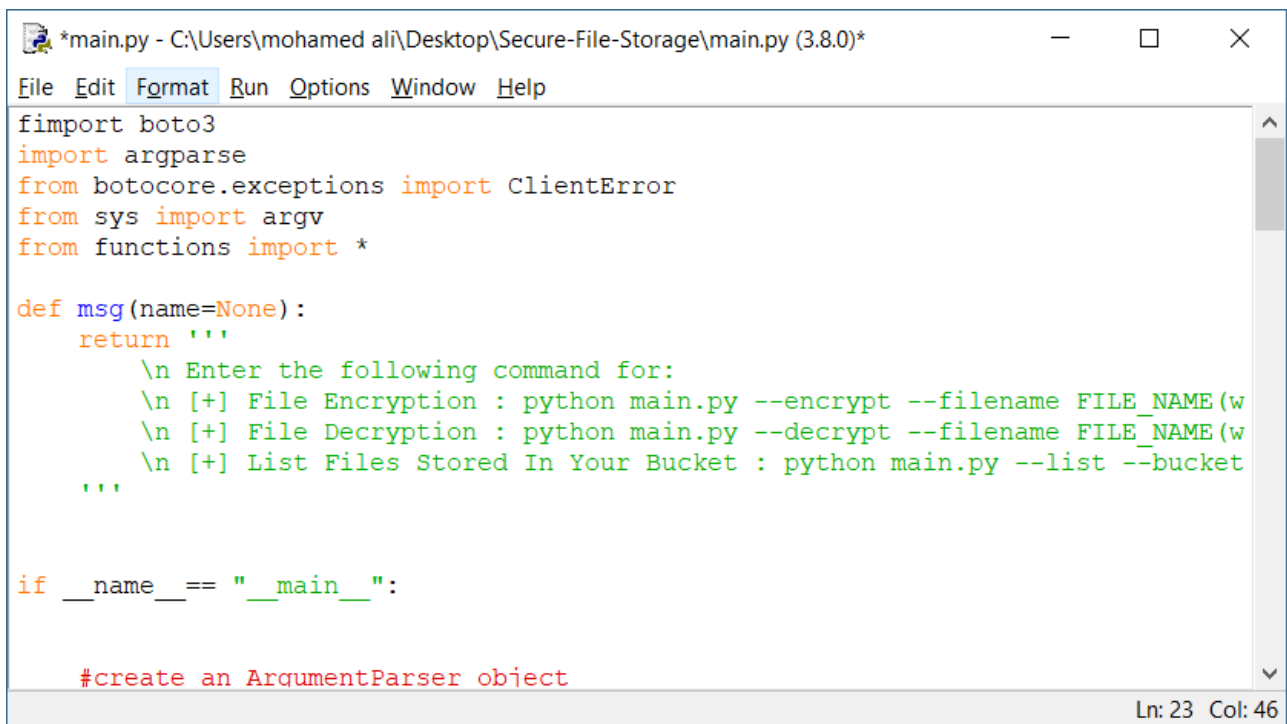
```

2. **Using IDLE Shell:** when a user install python, it comes with a shell called IDLE, a user can use this shell to run our program as follows:
  - (a) when a user opens idle it comes with the following interface



(b) from file à open à Choose the program directory à Choose main.py file





```

*main.py - C:\Users\mohamed ali\Desktop\Secure-File-Storage\main.py (3.8.0)*
File Edit Format Run Options Window Help
fimport boto3
import argparse
from botocore.exceptions import ClientError
from sys import argv
from functions import *

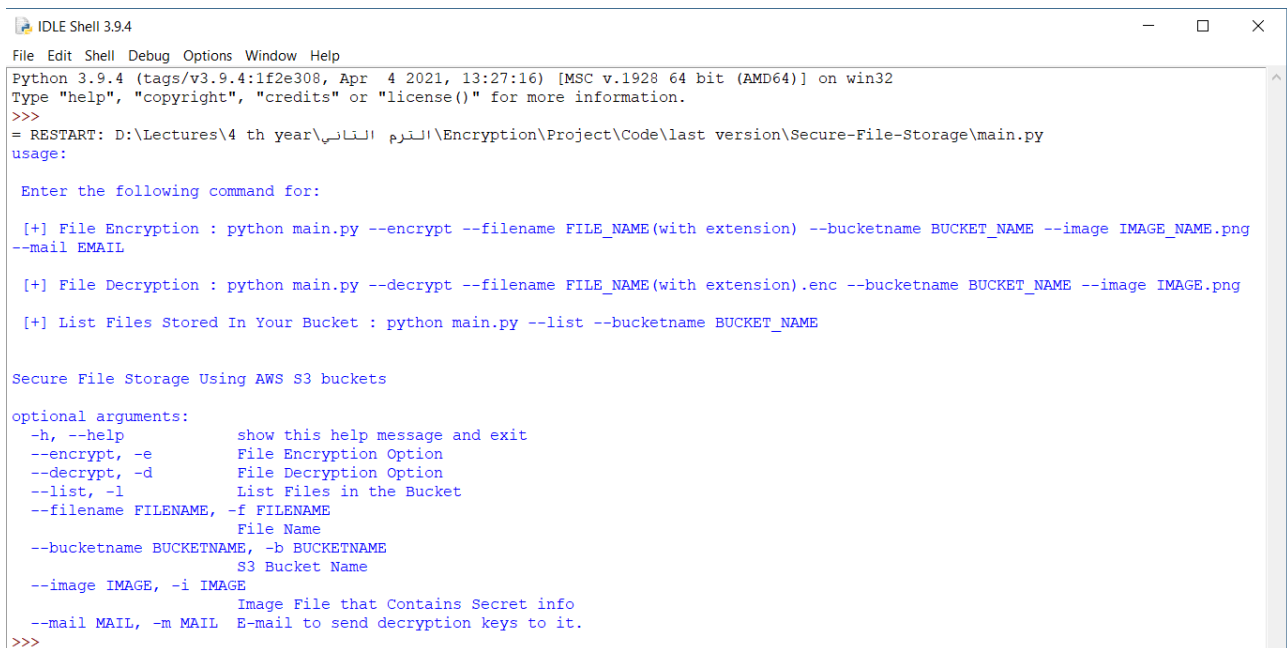
def msg(name=None):
    return '''
        \n Enter the following command for:
        \n [+] File Encryption : python main.py --encrypt --filename FILE_NAME(w
        \n [+] File Decryption : python main.py --decrypt --filename FILE_NAME(w
        \n [+] List Files Stored In Your Bucket : python main.py --list --bucket
    '''

if __name__ == "__main__":

    #create an ArgumentParser object
  
```

Ln: 23 Col: 46

the main.py will be opened, then from run choose run module:



```

IDLE Shell 3.9.4
File Edit Shell Debug Options Window Help
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 4 2021, 13:27:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\Lectures\4 th year\التاسع\Encryption\Project\Code\last version\Secure-File-Storage\main.py
usage:

Enter the following command for:

[+] File Encryption : python main.py --encrypt --filename FILE_NAME(with extension) --bucketname BUCKET_NAME --image IMAGE_NAME.png
--mail EMAIL

[+] File Decryption : python main.py --decrypt --filename FILE_NAME(with extension).enc --bucketname BUCKET_NAME --image IMAGE.png

[+] List Files Stored In Your Bucket : python main.py --list --bucketname BUCKET_NAME

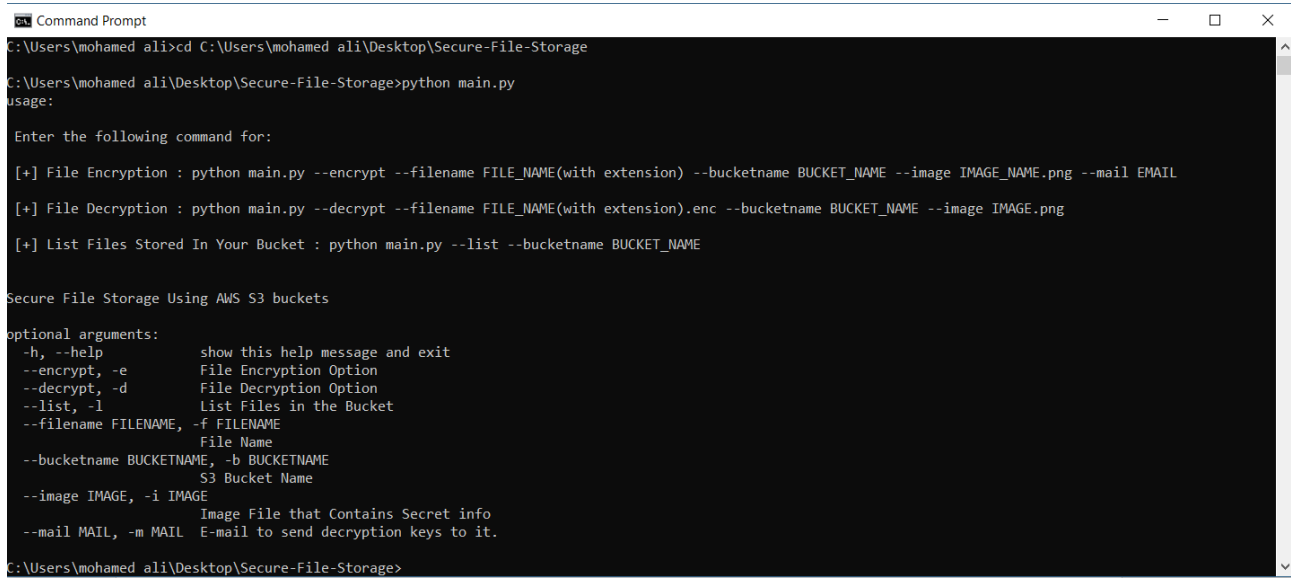
Secure File Storage Using AWS S3 buckets

optional arguments:
  -h, --help            show this help message and exit
  --encrypt, -e          File Encryption Option
  --decrypt, -d          File Decryption Option
  --list, -l            List Files in the Bucket
  --filename FILENAME, -f FILENAME
                        File Name
  --bucketname BUCKETNAME, -b BUCKETNAME
                        S3 Bucket Name
  --image IMAGE, -i IMAGE
                        Image File that Contains Secret info
  --mail MAIL, -m MAIL  E-mail to send decryption keys to it.
>>>
  
```

3. **Using IDE (Integrated Development Environment):** a user can run our program from any IDE that supports python such as PyCharm or Atom.

## 4.3 How a user can interact with our program

After a user run the program with one of the previous ways, the help menu and all available features are shown for him:



```

C:\Users\mohamed ali>cd C:\Users\mohamed ali\Desktop\Secure-File-Storage
C:\Users\mohamed ali\Desktop\Secure-File-Storage>python main.py
usage:

Enter the following command for:

[+] File Encryption : python main.py --encrypt --filename FILE_NAME(with extension) --bucketname BUCKET_NAME --image IMAGE_NAME.png --mail EMAIL
[+] File Decryption : python main.py --decrypt --filename FILE_NAME(with extension).enc --bucketname BUCKET_NAME --image IMAGE.png
[+] List Files Stored In Your Bucket : python main.py --list --bucketname BUCKET_NAME

Secure File Storage Using AWS S3 buckets

optional arguments:
  -h, --help            show this help message and exit
  --encrypt, -e          File Encryption Option
  --decrypt, -d          File Decryption Option
  --list, -l            List Files in the Bucket
  --filename FILENAME, -f FILENAME
                        File Name
  --bucketname BUCKETNAME, -b BUCKETNAME
                        S3 Bucket Name
  --image IMAGE, -i IMAGE
                        Image File that Contains Secret info
  --mail MAIL, -m MAIL  E-mail to send decryption keys to it.

C:\Users\mohamed ali\Desktop\Secure-File-Storage>

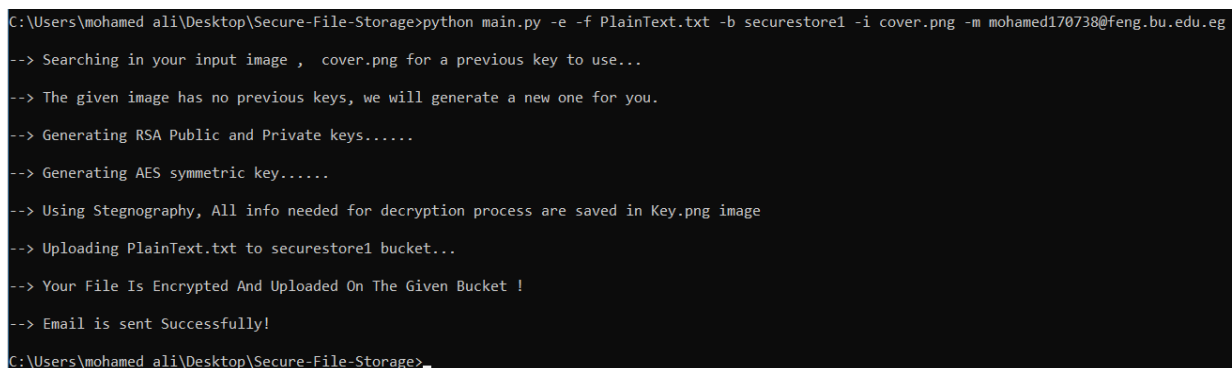
```

1. Show help menu
2. File Encryption.
3. File Decryption.
4. List Files stored in the bucket.

- **For File Encryption:**

**Example:** A user enters the following command in CMD:

*> python main.py -e -f PlainText.txt -b securestore1 -i cover.png -m mohamed170738@feng.bu.edu.eg*



```

C:\Users\mohamed ali\Desktop\Secure-File-Storage>python main.py -e -f PlainText.txt -b securestore1 -i cover.png -m mohamed170738@feng.bu.edu.eg
--> Searching in your input image , cover.png for a previous key to use...
--> The given image has no previous keys, we will generate a new one for you.
--> Generating RSA Public and Private keys.....
--> Generating AES symmetric key.....
--> Using Stegography, All info needed for decryption process are saved in Key.png image
--> Uploading PlainText.txt to securestore1 bucket...
--> Your File Is Encrypted And Uploaded On The Given Bucket !
--> Email is sent Successfully!

C:\Users\mohamed ali\Desktop\Secure-File-Storage>

```

To reduce the number of keys that the program produces, the first step that the program performs is searching in the cover image that the user enters for previous keys to use for encryption process, if there are not, the system generates new keys for this user.

- Before the encryption process the files in the bucket are:

**Objects (11)**

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Copy S3 URI](#)
[Copy URL](#)
[Download](#)
[Open](#)
[Delete](#)
[Actions](#)
[Create folder](#)
[Upload](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	attack.txt.enc	enc	April 22, 2022, 22:16:07 (UTC+02:00)	311.0 B	Standard
<input type="checkbox"/>	base.txt.enc	enc	April 20, 2022, 04:46:19 (UTC+02:00)	327.0 B	Standard
<input type="checkbox"/>	DSC_0027_4969.JPG	JPG	March 17, 2022, 21:54:40 (UTC+02:00)	2.9 MB	Standard
<input type="checkbox"/>	Example.txt.enc	enc	April 22, 2022, 20:12:17 (UTC+02:00)	375.0 B	Standard
<input type="checkbox"/>	first.txt	txt	April 11, 2022, 17:21:31 (UTC+02:00)	46.0 B	Standard
<input type="checkbox"/>	photo.jpg	jpg	April 11, 2022, 17:49:37 (UTC+02:00)	729.6 KB	Standard
<input type="checkbox"/>	requirements.txt.enc	enc	April 25, 2022, 04:43:30 (UTC+02:00)	343.0 B	Standard
<input type="checkbox"/>	Restful apis.pptx.enc	enc	April 22, 2022, 18:18:10 (UTC+02:00)	378.3 KB	Standard
<input type="checkbox"/>	song.mp3.enc	enc	April 22, 2022, 20:23:41 (UTC+02:00)	3.7 MB	Standard
<input type="checkbox"/>	to.txt.enc	enc	April 20, 2022, 05:06:30 (UTC+02:00)	311.0 B	Standard
<input type="checkbox"/>	USER.png.enc	enc	April 25, 2022, 01:52:54 (UTC+02:00)	2.5 KB	Standard

- After encryption process is done successfully the encrypted file is stored in the bucket as follow:

**Objects (12)**

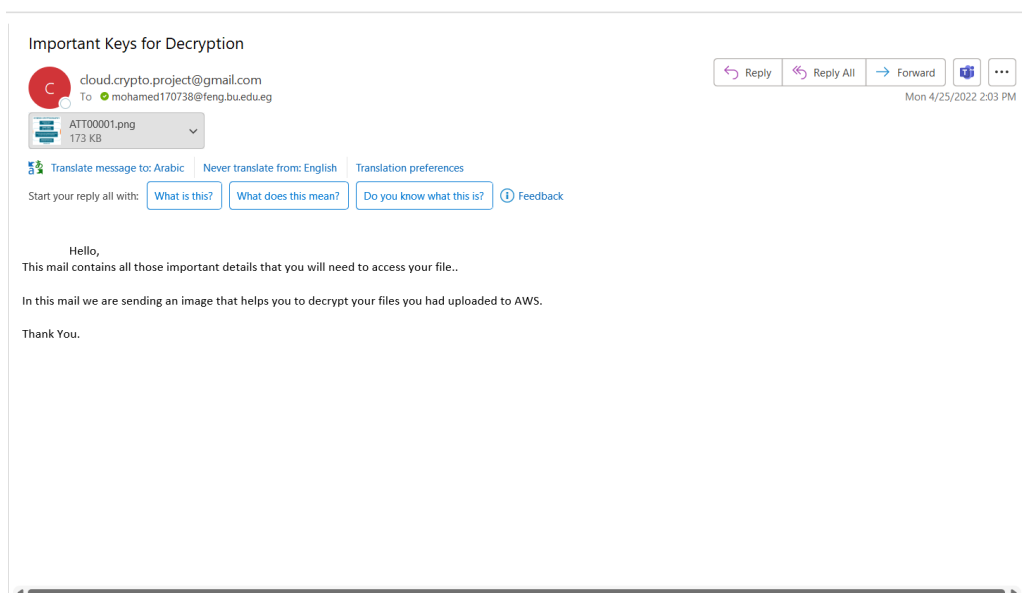
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Copy S3 URI](#)
[Copy URL](#)
[Download](#)
[Open](#)
[Delete](#)
[Actions](#)
[Create folder](#)
[Upload](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	attack.txt.enc	enc	April 22, 2022, 22:16:07 (UTC+02:00)	311.0 B	Standard
<input type="checkbox"/>	base.txt.enc	enc	April 20, 2022, 04:46:19 (UTC+02:00)	327.0 B	Standard
<input type="checkbox"/>	DSC_0027_4969.JPG	JPG	March 17, 2022, 21:54:40 (UTC+02:00)	2.9 MB	Standard
<input type="checkbox"/>	Example.txt.enc	enc	April 22, 2022, 20:12:17 (UTC+02:00)	375.0 B	Standard
<input type="checkbox"/>	first.txt	txt	April 11, 2022, 17:21:31 (UTC+02:00)	46.0 B	Standard
<input type="checkbox"/>	photo.jpg	jpg	April 11, 2022, 17:49:37 (UTC+02:00)	729.6 KB	Standard
<input checked="" type="checkbox"/>	PlainText.txt.enc	enc	April 25, 2022, 14:03:02 (UTC+02:00)	2.4 KB	Standard
<input type="checkbox"/>	requirements.txt.enc	enc	April 25, 2022, 04:43:30 (UTC+02:00)	343.0 B	Standard
<input type="checkbox"/>	Restful apis.pptx.enc	enc	April 22, 2022, 18:18:10 (UTC+02:00)	378.3 KB	Standard
<input type="checkbox"/>	song.mp3.enc	enc	April 22, 2022, 20:23:41 (UTC+02:00)	3.7 MB	Standard
<input type="checkbox"/>	to.txt.enc	enc	April 20, 2022, 05:06:30 (UTC+02:00)	311.0 B	Standard
<input type="checkbox"/>	USER.png.enc	enc	April 25, 2022, 01:52:54 (UTC+02:00)	2.5 KB	Standard

- The email that contains the stego image that contains the private key, and the encrypted symmetric key is sent from the main email of the program which is “cloud.crypto.project@gmail.com” to the email that the user enters as follows:



- **For File Decryption:**

After the user got the key.png image from the email he/she can download the encrypted file and decrypt it with.

**Example:** a user enters the following command in CMD :

> *python main.py -d -f PlainText.txt.enc -b securestore1 -i key.png*

```

C:\Users\mohamed ali\Desktop\Secure-File-Storage>python main.py -d -f PlainText.txt.enc -b securestore1 -i key.png
--> Working on Decrypting your file.....
--> File is Decrypted Successfully!
C:\Users\mohamed ali\Desktop\Secure-File-Storage>

```

### Before Decryption

Name	Date modified	Type	Size
.git	4/25/2022 12:56 PM	File folder	
__pycache__	4/25/2022 2:32 PM	File folder	
commands for trials.txt	4/25/2022 4:44 AM	Text Document	1 KB
Cover.png	3/6/2022 1:16 AM	PNG File	113 KB
flag.txt	4/24/2022 6:12 PM	Text Document	1 KB
functions.py	4/25/2022 2:30 PM	Python File	5 KB
info.txt	4/24/2022 6:12 PM	Text Document	11 KB
key.png	4/25/2022 2:02 PM	PNG File	128 KB
main.py	4/25/2022 12:30 PM	Python File	8 KB
README.md	4/24/2022 6:12 PM	MD File	1 KB
requirements.txt	4/24/2022 6:12 PM	Text Document	1 KB

### After Decryption

Name	Date modified	Type	Size
.git	4/25/2022 12:56 PM	File folder	
__pycache__	4/25/2022 2:32 PM	File folder	
commands for trials.txt	4/25/2022 4:44 AM	Text Document	1 KB
Cover.png	3/6/2022 1:16 AM	PNG File	113 KB
flag.txt	4/24/2022 6:12 PM	Text Document	1 KB
functions.py	4/25/2022 2:30 PM	Python File	5 KB
info.txt	4/24/2022 6:12 PM	Text Document	11 KB
key.png	4/25/2022 2:02 PM	PNG File	128 KB
main.py	4/25/2022 12:30 PM	Python File	8 KB
PlainText.txt	4/25/2022 2:39 PM	Text Document	3 KB
PlainText.txt.enc	4/25/2022 2:39 PM	Wireshark capture ...	3 KB
README.md	4/24/2022 6:12 PM	MD File	1 KB
requirements.txt	4/24/2022 6:12 PM	Text Document	1 KB

- **For Listing Stored Files in AWS S3 bucket:**

**Example:** a user enters the following command: *python main.py -l -b securestore1*

```

C:\Users\mohamed ali\Desktop\Secure-File-Storage>python main.py -l -b securestore1
[+] DSC_0027_4969.JPG
[+] Example.txt.enc
[+] PlainText.txt.enc
[+] Restful apis.pptx.enc
[+] USER.png.enc
[+] attack.txt.enc
[+] base.txt.enc
[+] first.txt
[+] photo.jpg
[+] requirements.txt.enc
[+] song.mp3.enc
[+] to.txt.enc
C:\Users\mohamed ali\Desktop\Secure-File-Storage>_

```

## 4.4 How a user can handle any errors due to entering a wrong argument

Errors are produced to the following reasons:

- **During Encryption:**

1. A user does not enter all arguments needed for the encryption command.
2. A user enters a filename that is not found.
3. A user enters a cover image name that is not found.
4. A user enters a wrong bucket name.
5. A user enters an invalid email.

- **During Decryption:**

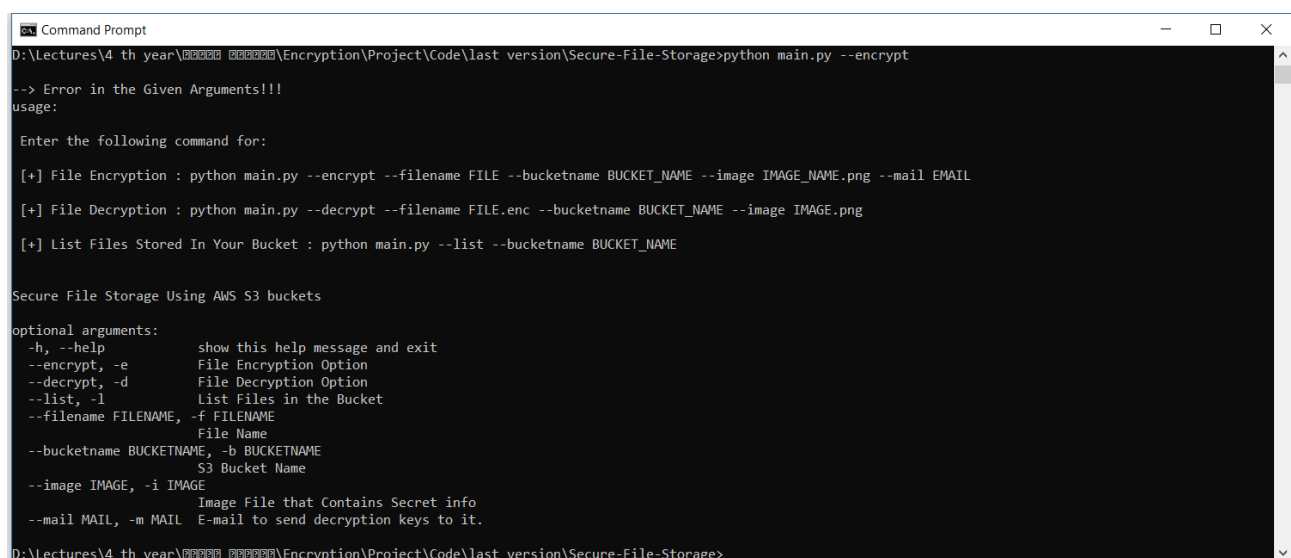
1. A user does not enter all arguments needed for the decryption command.
2. A user enters an encrypted filename that is not found.
3. A user enters an image that does contain decryption key.
4. A user enters a wrong bucket name.

- **Listing Stored file in S3 bucket:**

1. A user does not enter all arguments needed for the decryption command.
2. A user enters wrong bucket name.

So, in all these error cases a help message appears for the user to tell him the error and tell him that he/she enter all the arguments in the write way.

### Example:



```

Command Prompt
D:\Lectures\4 th year\Encryption\Project\Code\last version\Secure-File-Storage>python main.py --encrypt
--> Error in the Given Arguments!!!
usage:
Enter the following command for:
[+] File Encryption : python main.py --encrypt --filename FILE --bucketname BUCKET_NAME --image IMAGE_NAME.png --mail EMAIL
[+] File Decryption : python main.py --decrypt --filename FILE.enc --bucketname BUCKET_NAME --image IMAGE.png
[+] List Files Stored In Your Bucket : python main.py --list --bucketname BUCKET_NAME

Secure File Storage Using AWS S3 buckets
optional arguments:
-h, --help            show this help message and exit
--encrypt, -e          File Encryption Option
--decrypt, -d          File Decryption Option
--list, -l            List Files in the Bucket
--filename FILENAME, -f FILENAME
                        File Name
--bucketname BUCKETNAME, -b BUCKETNAME
                        S3 Bucket Name
--image IMAGE, -i IMAGE
                        Image File that Contains Secret info
--mail MAIL, -m MAIL  E-mail to send decryption keys to it.
D:\Lectures\4 th year\Encryption\Project\Code\last version\Secure-File-Storage>

```



# Bibliography

- [1] Amazon s3 - object storage built to retrieve any amount of data from anywhere. [https://aws.amazon.com/s3/?nc1=h\\_ls](https://aws.amazon.com/s3/?nc1=h_ls). Accessed: April 2022.
- [2] Create a bucket on amazon s3. <https://docs.aws.amazon.com/AmazonS3/latest/userguide/creating-bucket.html>. Accessed: April 2022.
- [3] Create account on amazon. <https://aws.amazon.com/>. Accessed: April 2022.
- [4] Cryptography. <https://en.wikipedia.org/wiki/Cryptography>. Accessed: March 2022.
- [5] Python packages. <https://www.python.org/downloads/>. Accessed: April 2022.
- [6] Setup amazon s3 bucket using python. <https://towardsdatascience.com/how-to-upload-and-download-files-from-aws-s3-using-python-2022-4c9b787b15f2>. Accessed: April 2022.
- [7] Steganography. <https://en.wikipedia.org/wiki/Steganography>. Accessed: March 2022.
- [8] Symmetric vs. asymmetric encryption. <https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences>. Accessed: March 2022.
- [9] What is the cloud? | cloud definition. <https://www.cloudflare.com/>. Accessed: March 2022.
- [10] Image steganography. <https://www.geeksforgeeks.org/image-steganography-in-cryptography/>, 2021. Accessed: March 2022.
- [11] Steganography. <https://www.simplilearn.com/what-is-steganography-article>, 2022. Accessed: March 2022.
- [12] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2020.
- [13] Casey Clark Margie Semilof. Steganography. <https://www.techtarget.com/searchsecurity/definition/steganography>. Accessed: March 2022.
- [14] Douglas R Stinson. *Cryptography: theory and practice*. Chapman and Hall/CRC, 2005.