



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

DATA PROTECTION AND ENCYPTION SECURITY ON CLOUD

Name: Bobbala Mahith kumar reddy | Reg no: 18BIT0076 |

Name: Nalla sanjay reddy | Reg no: 18BIT0100 |

Faculty: Prof SUMAIYA THASEEN | CSE3502 - Information Security Management

Google drive link:

https://drive.google.com/drive/folders/1qOniKuihitNVMft3mvM_Ds6XknbolZLD?usp=sharing

INTRO AND ABSTRACT

Our project deals with an Enhanced cloud storage system for Dropbox and Google Drive.

We will implement a software utility that enhances a user's privacy when storing and sending data files in cloud storage, specifically in DropBox and GoogleDrive using cryptographic algorithms . The software accomplishes the following goals:

- Encrypting a file, prior to being uploaded to the cloud storage
- Decrypting the encrypted file upon downloading
- Sharing the file with other users in a secure manner.

SO WHAT PROBLEM DOES OUR PROJECT SOLVES

1. If the credentials are stolen for dropbox or google, or if the device is lost or stolen , then the hacker can gain access to to all the files and sensitive data that are stored in the cloud. So by using our software we are encrypting the files and directly storing then on to the respective cloud platforms through the software by using the respective API access of the cloud. So even if credentials are lost or device is stolen, the hacker cannot read the sensitive data on the cloud.
- 2 . If a hacker is able to get the privileges of admin of the cloud service provider, then he can read all the data in the cloud even if the data is encrypted because he can get access to the keys stored seperately from admin privileges , and he can decrypt the data. So if we encrypt the files through our software before storing , we wil have our own private key with us , so even if hacker gets our data ,he cannot read it.

More on our project

There are two forms of encryption used in cloud storage services: transit and resting. The files and directories you upload to a cloud service are normally encrypted as they travel between your PC and the cloud server, using at least 128-bit secure sockets layer (SSL) technology.

When your data is saved, however, the chances of it being encrypted are slim, and if it is, the cloud provider will almost certainly have the key. Resting encryption is only available to OneDrive users with a corporate subscription. Dropbox, on the other hand, uses 256-bit encryption at rest, but Dropbox keeps the encryption keys.

Brute-force attacks are resisted in the cloud by encrypting data in transit and at rest. A supercomputer would take years to crack the 256-bit encryption. User failure is significantly

more likely to occur as a result of a phishing attack or a weak password that may be guessed by an interested party. If your account can be accessed through the front door, decryption isn't required.

Keeper of the encryption keys

Otherwise, a key is required to decrypt the encrypted data. Each time data is encrypted, one of these keys is generated and preserved. Many online backup solutions, as opposed to cloud-sync systems, allow you to create your own encryption key and take responsibility for its security. If you lose the key, your backed-up data will be lost forever.

Dropbox and other syncing and sharing cloud storage providers do not allow users to produce their own encryption keys. The user must put his or her trust in the service when it comes to the creation and storage of the key. You're less likely to get shut out permanently because you misplaced your card, but there's also the possibility that the service will be hacked, in which case you'll be out of luck. Bottom line: No matter how careful you are, things will go wrong when someone else has the keys.

Literature review

Colombo, M., Asal, R., Hieu, Q. H., El-Moussa, F. A., Sajjad, A., & Dimitrakos, T. (2019, July). Data protection as a service in the multi-cloud environment. In 2019 IEEE 12th International Conference on Cloud Computing (CLOUD) (pp. 81-85). IEEE.

This paper introduces a data protection framework as a service (DPaaS) to obscure computer users. Compared to existing data encryption (DEaaS) such as those provided by Amazon and Google, our DPaaS framework provides flexibility, control and visibility to protect data in the cloud. In addition to supporting basic data encryption as DEaaS does, this DPaaS framework allows data owners to define access control policies that are well-designed to protect their data.. Information protected by access control policy is encrypted automatically and access is granted to user / applications in accordance with policy. In general, DPaaS enables the separation of concerns between data security and management, in addition to defining the full automatic cycle of data security from encryption to encryption. The DPaaS concept model works with hybrid cloud compounds that include private cloud and virtual data centers using OpenStack, CloudStack and VMWare as well as public clouds on the BT Cloud Compute and Amazon (AWS) platform. Examination of the type used has proven the effectiveness of the framework.

Gadekar, D. P., Sable, N. P., & Raut, A. H. Exploring Data Security Scheme into Cloud Using Encryption Algorithms. International Journal of Recent Technology and Engineering (IJRTE), Published By: Blue Eyes Intelligence Engineering & Sciences Publication, ISSN, 2277-3878.

In this research paper they explained that using an algorithmic encryption system between a digital signature on a cloud-bound server. If a client uploads files to a cloud server intended for distribution to multiple clients, the closest should be a way to accomplish the file founder. Certification certification methods help to ensure the file developer has defined a PEKS image that provides security alongside three attacks. This attack is selected for keyword abuse, selected cipher attack and keyword attack And this image protects inside and outside the rival. Therefore, important security considerations (IND-SCF-CKCA and IND-KGA) are preprogrammed and defined as a method known as user authentication. According to this method, the information file is first divided into equal parts by the information vendor. After splitting the AES into each chunk is done. One-time encryption completed, with each part of the hash code created. Following the execution of all the steps, the encrypted enclosure is hosted in the cloud. While downloading a file, the client must proceed to the cloud to generate hash code for that file. To verify the authenticity of the information the client must verify the hash code and its hash code. In general tasks that come with the customer side and creating hash code and collecting encrypted data, the cloud is involved.

Sudha, I., & Nedunchelian, R. (2019). A secure data protection technique for healthcare data in the cloud using homomorphic encryption and Jaya-Whale optimization algorithm. International Journal of Modeling, Simulation, and Scientific Computing, 10(06), 1950040.

In this research paper they explained how managing big data sets and security are major problems in cloud systems Therefore, this paper proposes a process, namely, Jaya - Whale Optimization (JWO), which is a combination of Jaya algorithm and Whale optimization algorithm (WOA) and synchronizes encryption homomorphic to initiate secure data transfer in the cloud. The actual data is stored by generating a Data Protection (DP) coefficient using the proposed JWO algorithm. In the proposed algorithm, robustness is calculated according to the confidentiality criteria and the use of the appropriate solution. Also, refined information is generated by DISPOSAL Key Information Product (KIP) matrix and key vector. Finally, the data owner provides the key for users to access the original data in the refined data. Tests performed using Cleveland, Hungarian, and Swiss data sets in terms of BD, accuracy, robustness and analysis show that the proposed JWO provides high performance in terms of BD, accuracy, and rigidity parameters of 0.720, 0.822, and 0.722 values.

Sajay, K. R., Babu, S. S., & Vijayalakshmi, Y. (2019). Enhancing the security of cloud data using hybrid encryption algorithm. *Journal of Ambient Intelligence and Humanized Computing*, 1-10.

In this research paper they explained that the rapid growth of cloud computing technology and the rapid rise of cloud services became a major issue where data security is a challenging issue. The author lists all performance as long as security is in the cloud file. Eight encryption algorithms such as DES, AES, RC4, 3DES, RC6, Two-Fish and Blow-Fish are basic mathematical tests (NIST) and Pseudo Random Range Generator (PRNG). To select one of these algorithms, one software is required which is the required software for the sanctuary. The encryption level has the ability to calculate the performance of the algorithm. These eight algorithms are tested and this test is based on the P value and the negative response rate. They came up with a hybrid algorithm to improve cloud data security using encryption algorithm that combines homographic encryption with blowfish encryption to improve cloud security.

Qiu, H., Noura, H., Qiu, M., Ming, Z., & Memmi, G. (2019). A user-centric data protection method for cloud storage based on invertible DWT. *IEEE Transactions on Cloud Computing*.

In this paper, they have introduced a novel data protection approach that combines the concept of Selective Encryption (SE) with fragmentation and distribution in storage. This approach is based on the invertible Discrete Wavelet Transform (DWT) for dividing agnostic data into three pieces with three levels of protection. After that, these three pieces are scattered across different storage areas with different levels of reliability to protect end-user data against potential leaks in Clouds. Therefore, this approach increases the cost of storage by maintaining expensive, independent, and secure storage spaces and using a cheap but reliable storage space. have a high level of safety analysis done to ensure a high level of security of our approach. In addition, efficiency is evidenced by the implementation of the transfer of functions between the CPU and the General Purpose Graphic Processing Unit (GPGPU) in a structured manner.

Jakimoski, K. (2016). Security techniques for data protection in cloud computing. *International Journal of Grid and Distributed Computing*, 9(1), 49-56.-

In this paper they explained that cloud computing includes a private cloud, a hybrid cloud, and a public cloud. Types of service delivery, on the other hand, can be divided into SaaS (Software as a service), PaaS (Platform as a service), and IaaS (Infrastructure as a service). With cloud access, cloud computing is often categorized: public cloud (where the infrastructure is managed is a cloud vendor); private cloud (where computer infrastructure is provided by an organization and can be shared with other organizations); mixed cloud (the use of private and public clouds together); and social cloud (including the allocation of IT infrastructure between organizations of the same community). If this category depends on the type of services provided, the existing cloud is divided into the following categories: IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and Software as a Service (SaaS). data security challenges where cloud computing needs to be properly resolved and downgraded accordingly. In addition to the above, known attacks include phishing scams, IP spoofing, messaging, traffic analysis, IP ports, etc. There are many ways to protect data protection hosted by cloud computing providers, and all offer

authenticity, privacy, access control and authorization. This paper examines the feasibility of using an encryption algorithm for data security and privacy in Cloud Storage. Define requirements such as: Creating a system that will provide security and privacy in cloud storage, establishing an encrypted system based on cloud-sensitive data protection and Structure of how the service provider and storage will work with encrypted data, so Create a system where the user stores his data in the cloud on a computer where the user logs in to the cloud and stores data on the storage device. The server's cloud storage is not as secure as it can be read by anyone with permission to log in and leave the data at risk, Creating a retrieval system where the user gets encrypted and deleted by the user. and privacy for both buttons that run at user level

Sarmila, K. B., & Manisekaran, S. V. (2019, October). A Study on Security Considerations in IoT Environment and Data Protection Methodologies for Communication in Cloud Computing. In 2019 International Carnahan Conference on Security Technology (ICCST) (pp. 1-6). IEEE.

This paper demonstrates the impact and importance of cloud computing on IoT, as large data bytes made on IoT devices require a proper data acquisition system. IoT is used in a variety of fields such as industry, medicine, automotive and many other applications to simplify people's lives. As technology is used in various applications, threats to IoT are also increasing. For secure communication on IoT, there is a need for various levels of security algorithms (such as encryption techniques, etc.). work on analytical testing of data protection methods. The discussion of the environment and the importance of Lightweight writing techniques suitable for the IoT environment are discussed. This paper also highlights the need for End to End Encryption in the IoT environment. Signal protocol defined by a security protocol that provides applications with end-to-end encryption. This paper concludes with several guidelines for future research on safety considerations and solutions.

MODULES DESCRIPTION AND IMPLEMENTATION:

Pycrypto:

PyCrypto is a python encryption library that may be used for almost anything. The Paramiko SSH module for Python is an example of this, as it uses PyCrypto as a dependency to encrypt data. PyCrypto is a Python-based encryption library that is simple to use but incredibly powerful and helpful. We can have built-in functions for DES, AES, RSA, and other algorithms. We can encrypt an info into a cipher using AES using `EncodeAES(cipher,info)`, and we can decrypt an encrypted cipher using `DecodeAES(cipher,encryption)`.

Dropbox:

Dropbox for Python is a built-in tool that allows you to easily integrate DropBox into your Python program. You'll need to create a new app in the App Console to use the DropBox API. We must first choose the Dropbox API app and then the app's permissions. To access API v2, we'll need to use the app key created with this app. An instance of the Dropbox object is required in order to make API requests. To begin, enter the access token for the account you want to link, or use the App Console to get an access token for our own account.

Dropbox is abbreviated as dbx.

We utilize `dropbox.oauth` to authorize the app console and we utilize `Dropbox('OUR ACCESS TOKEN')`.

For authorizing using key and secret, use `DropboxOAuth2FlowNoRedirect`. We'll define a key and secret to the app console in the code, and we'll generate paths for the files and keys using the authorization code result to authorize the application, and in the case of file upload, we'll generate RSA key pairs for the client, then the file by reading and padding and generating SHA256 hashed stringContent as secretkey, and then it'll encrypt the secretkey.

Desktop app on Secure file storage and transfer in google drive

PROJECT GOAL

The project's main purpose is to create an application that connects to an online cloud storage platform and can upload data while encrypting them, and then decode the encrypted file as you get it. The program should be able to handle numerous users logging in and downloading these encrypted files, as well as sharing the key among them.

Research

Because we decided to write the software in Python, we focused much of my study on Python libraries and APIs.

We had to first decide which cloud storage platform we wanted to use. we didn't have to do much research because we are a frequent user of cloud storage platforms and have settled on Google Drive. we then had to investigate the APIs that Google offers for drive. Fortunately, they have a comprehensive QuickStart Guide¹ that showed me how to authenticate the user connecting to the cloud service. There was also a guide for uploading files, downloading files, and searching for files, so the guide did not end there. There were more guides, but these were sufficient for us to create the required program.

We then looked into cryptography for Python and discovered a popular library that does just that. Cryptography is the name of the library I'm using for the cryptography portion of the assignment, and it, too, has excellent documentation. Fernet is the name of the encryption I'll be using; it uses AES in CBC mode with a 128-bit key for encryption and HMAC with SHA256 for authentication. It uses the computer's salt to initialize vectors.

Next, we looked into common methods for storing credentials such as username and password and discovered pickle, a tool that allows you to save Python objects and re-load them into the program after exiting. This meant that we could use any username and password system at any time. A key-value pair (dictionary or hash map) is a common solution for this.

The classes we made are:

- main.py
- users.py
- googleDriveAPI.py
- menuTypes.py
- encryption.py
- auth.py

The program's main line will be Main.py. This is what will be used to generate the user's front end. It's short and simple, and it uses the user class for login and the menuType class for menu generation.

Users.py is where user names and passwords are created, erased, and saved, as the name implies.

GoogleDriveAPI.py is the interface between my software and Google Drive, and it will allow me to retrieve and upload files as needed.

MenuTypes.py is the class that builds menus for both privileged (admin) and non-privileged (ordinary) users. It uses the majority of classes to perform the essential functions.

Encryption.py, as the name implies, is the class that performs encryption and decryption operations as well as key generation.

When calling the APIs, Auth.py is utilized to authenticate the Google Drive user.

Main.py

main ()

When the application initially starts, it checks to see if a key exists; if it does not, a new one is generated immediately. The application then checks to see if any previous users have registered. If there are no users, the user is prompted to create an admin account, which has the ability to create new accounts and produce new keys.

Otherwise, the user would be prompted to log in. If you log in as an administrator, you'll get a menu with additional features tailored to that user. If you don't, you'll be provided with a regular menu. This runs indefinitely to allow people to log in and out whenever they want.

Users.py

Save_list(obj, name)

This function is used to pickle (store) the list of users that are currently participating in the program. I store the user name and password as a key value pair in a dictionary. It is invoked whenever a new user is created.

load_list(name)

This function loads the object that contains the users and passwords. When the user class is called, it checks to see if the usernames are correct.

newUser()

To create a new user, use this function. It's straightforward: it asks for a user name, alerts the user if the user name already exists, and then asks for a password, which it puts in our dictionary and pickles.

Login()

This function is called on the main line and displays a login page to the user. It checks if the user connecting in is an admin, an ordinary user, or if they fail to login three times, the session is terminated.

`deleteUser()`

The administrator uses this to delete a user. It prevents the admin account from being deleted and notifies the user if the user does not exist. When a user is deleted, the username dictionary is pickled and safely saved once more.

GoogleDriveAPI.py

Here there are all relevant functions required to carry out uploading downloading and filesearching. The functions include:

- `uploadFile(fileName)`
- `downloadFile(file_id, fileName)`
- `searchFile(query)`
- `fileID(query)`

MenuTypes.py

`PrivilegeMenu()`

This creates a menu for the administrator that includes extra options like adding and deleting users as well as producing a new key. The majority of activities are conducted here, such as uploading a file, which will be encrypted before being uploaded, and downloading a file from the disk, which will be downloaded and decrypted.

`StandardMenu()`

This menu is nearly identical to the above menu however the settings option is removed to prevent the user from performing privileged actions.

Encryption.py

`KeyGen()`

`KeyGen` as the name suggests is used to generate a new key and store it on the computer.

`KeyRead()`

This is used to read the key from the computer, if no key exists it will automatically generate a new one by calling `keyGen()`.

`Encrypt(filename, Key)`

This is used to encrypt the file, it is using Fernet to perform the required encryption.

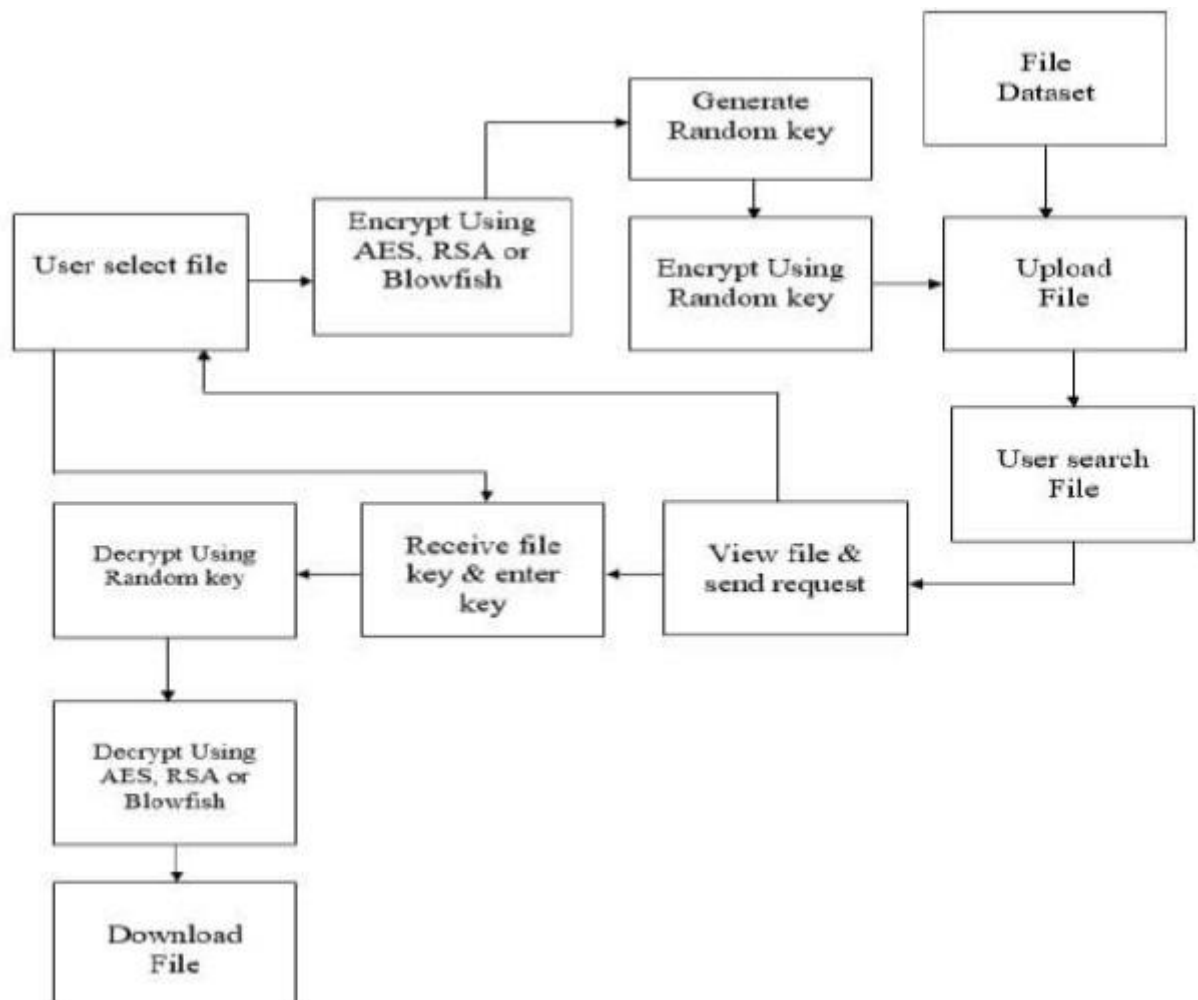
`Decrypt(filename, key)`

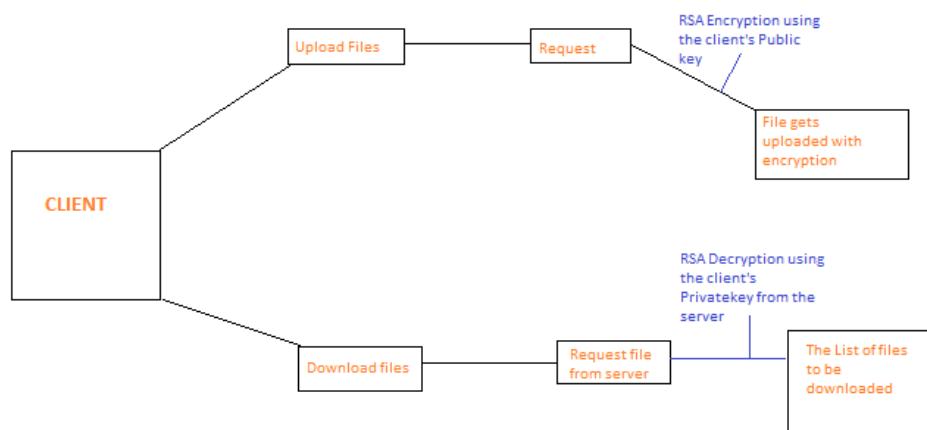
This is used to decrypt the file that had been encrypted.

Auth.py

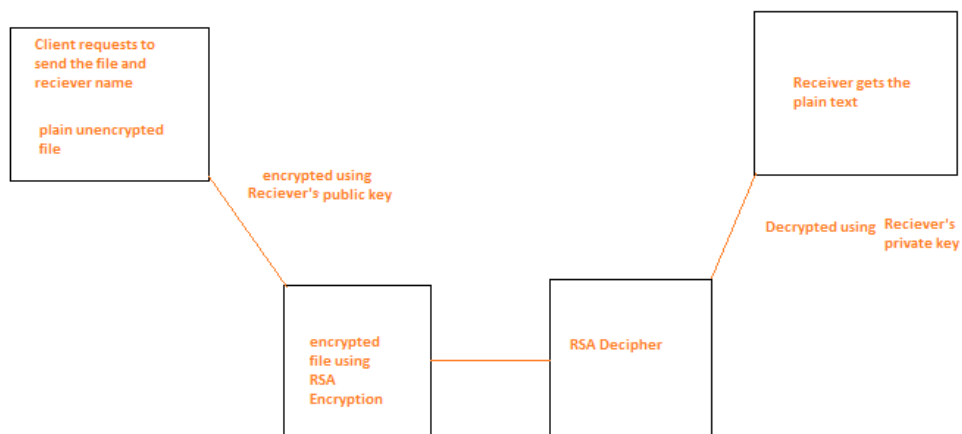
This is used to authenticate with Google Drive. This is taken from the Google Drive API QuickStart guide.

Review 2 implementation:





File sharing among friends



Implementation output:

Running the main file:

```
* * * * *
```

```
AssureCloud : Secure data storage and privacy protection for Dropbox clients
```

```
* * * * *
```

```
Can you please authenticate yourself?
```

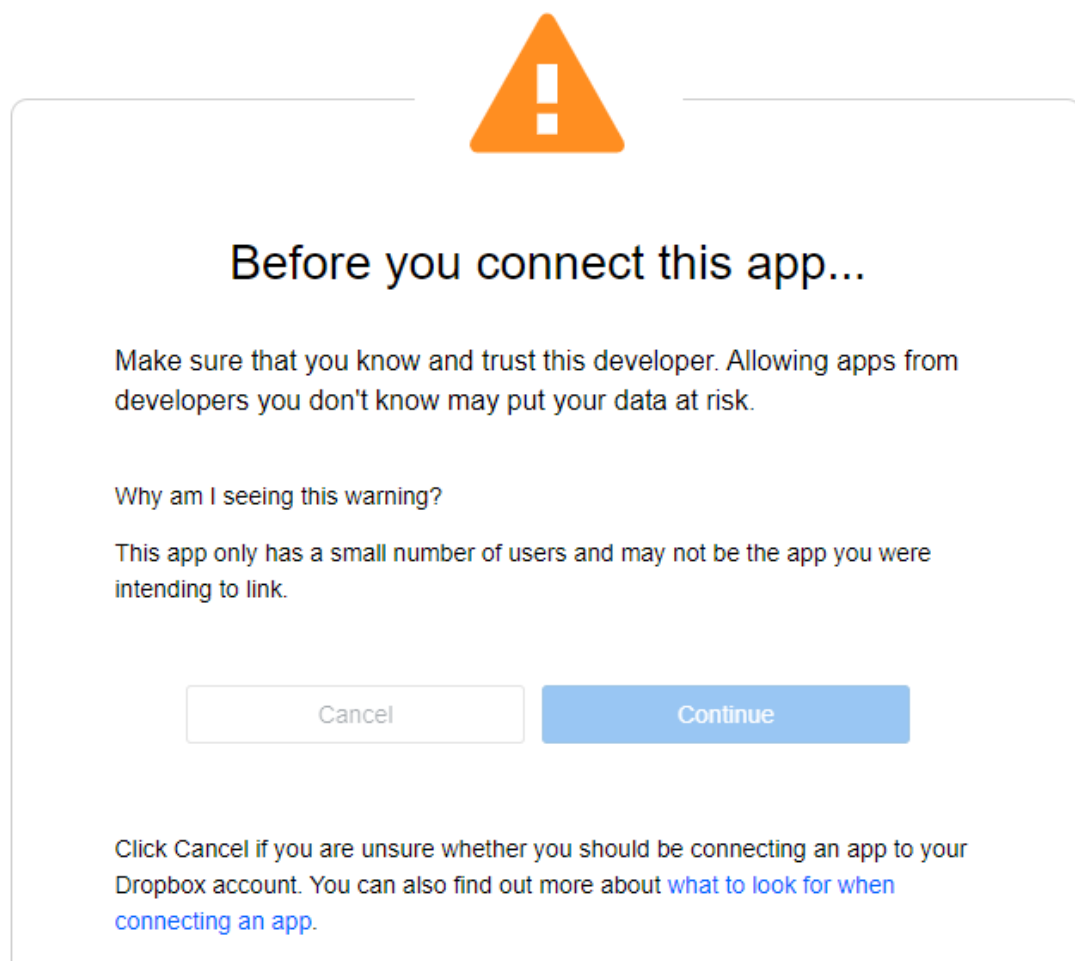
```
1. Go to: https://www.dropbox.com/oauth2/authorize?response_type=code&client_id=x03ls4fvmfws3v5
```

```
2. Click "Allow" (you might have to log in first)
```

```
3. Copy the authorization code
```

```
Enter the authorization code here:
```

Authorizing the App:





AssureCloud would like access to its own folder,
Apps > **AssureCloud**, inside your Dropbox. [Learn more](#)

Cancel

Allow



Enter this code into **AssureCloud** to finish the process.

t5Un8MKMDq4AAAAAAAAAMgzEwSBICokf6IaLMEWkFMQ

Entering the authorization code and user is authenticated successfully

```
Enter the authorization code here: t5Un8MKMDq4AAAAAAAAAMgzEwSB1Cokf6IaLMEWkFMQ
Authentication successful!

Generating RSA key pair for Mahith

Hello, Mahith!

What do you want to do next . . .
1. Upload a file
2. Download a file
3. Share the file with friend
4. Exit

Enter your choice here :
```

RSA Key Pair generated for the user:

This PC > Local Disk (G:) > academics > isaanascom > project > review 2 18BIT0076 18BIT0100 > dropbox cloud project > keys				
Name	Date modified	Type	Size	
Mahith_public_rsa_key.pem	18-05-2021 20:55	PEM File	1 KB	
Mahith_pvt_rsa_key.pem	18-05-2021 20:55	PEM File	1 KB	

Feature 1: Encryption of the file and uploading it to dropbox

Before uploading:

Dropbox > Apps > AssureCloud > data

Search

Overview

Hide

Upload

Create

...

Menu

This folder is empty
Drag and drop files into this window to upload.

File upload:

```
4. EXECUTING THE PROGRAM
Enter your choice here : 1

UPLOAD FILE FEATURE

Enter file name: sample.html
File encryption in progress . . .
Mahith, your file is encrypted successfully!

Creating new encrypted secret key

File upload in progress . . .
Mahith, your encrypted file has been successfully uploaded!
```


After uploading:

Dropbox > Apps > AssureCloud > data

Search

Overview

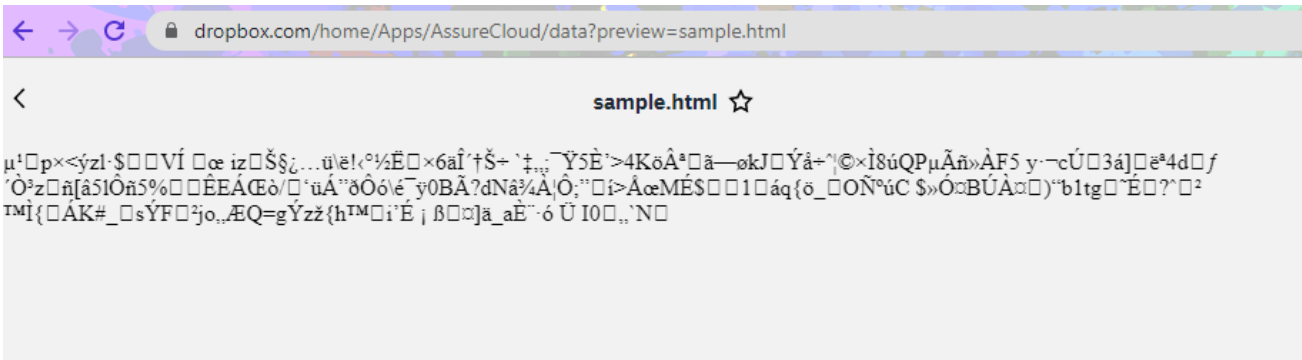
Hide

Click here to describe this folder and turn it into a Space

Show examples

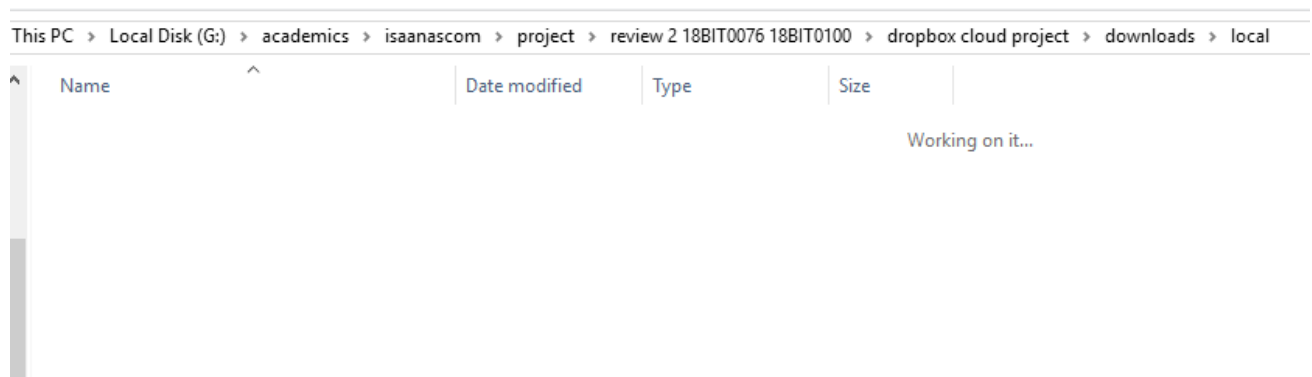
<div><div>Upload</div><div>Create</div><div></div></div>				
Name	Modified	Members		
<div><div></div>sample.html</div>	<div><div></div>Today at 21:00</div>	Only you		

File uploaded in an encrypted form:



Feature 2: Downloading the file and decrypting it to view it locally

Before downloading:




File download:

```
4. EXIT
Enter your choice here : 2

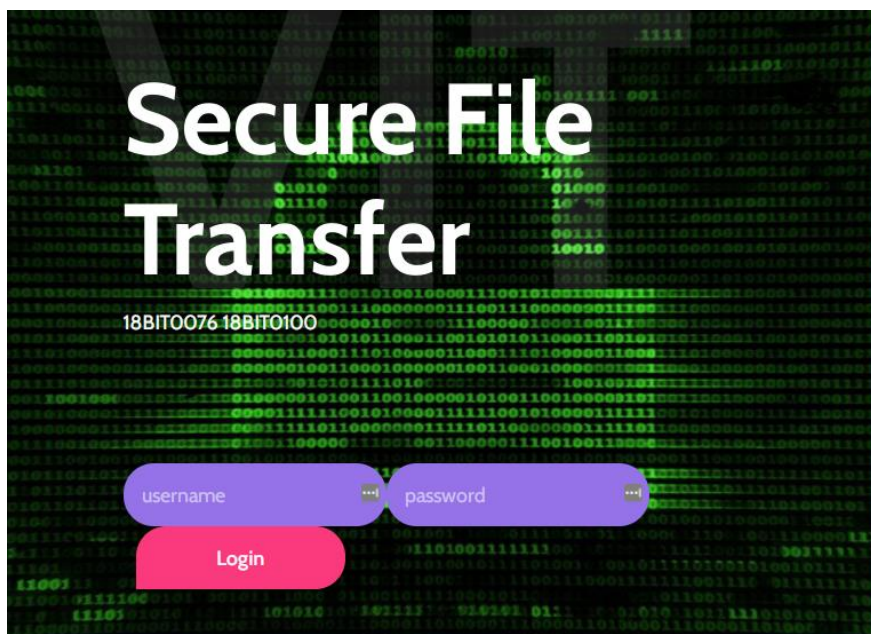
DOWNLOAD FILE FEATURE

Enter file name: sample.html
Downloading the file - sample.html
Download location - ../downloads/local
Download successfully complete!
```

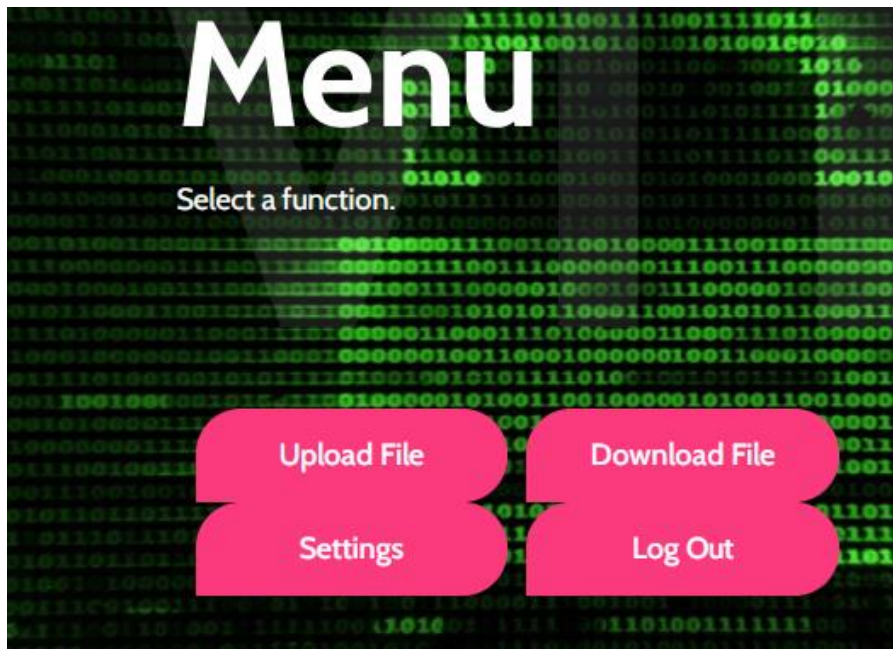
After download:

This PC > Local Disk (G:) > academics > isaanascom > project > review 2 18BIT0076 18BIT0100 > dropbox cloud project > downloads > local				
^	Name	Date modified	Type	Size
n	 sample.html	18-05-2021 21:05	Chrome HTML Do...	1 KB

Review 3 implementation snapshots: implementing GUI and peer to peer transfer of files through encryption in cloud

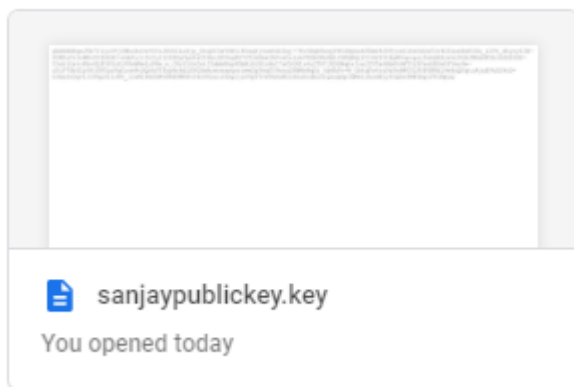


Main screen



Then upload of public key for the receiver to the sender

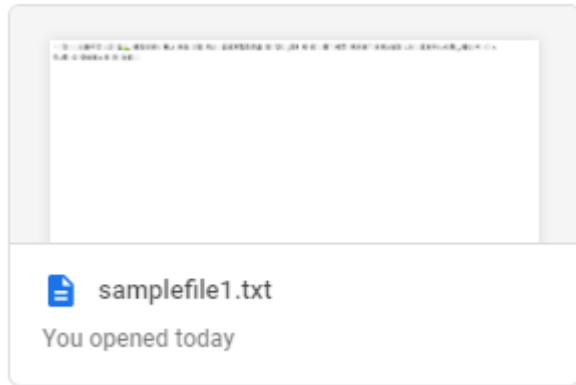




Then download of public key by sender



Then upload of file using public key of receiver



Then download of file by receiver



Performance analysis between RSA algorithm and AES algorithm on two different machines
two different machines: P-II 266 MHz and P-4 2.4 GHz.

Algorithm	Megabytes(2^{20} bytes) Processed	Time Taken	MB/Second
AES	256	2.976 ms	64.386
RSA	256	4.196 ms	41.010

Input Size (bytes)	RSA	AES
20,527	2	4
36,002	4	6
45,911	5	8
59,852	7	11
69,545	9	13
137,325	17	26
158,959	20	30
166,364	21	31
191,383	24	36
232,398	30	44
Average Time	14	21
Bytes/sec	7,988	5,320

From this info we can see that AES is computationally quicker than RSA

AES has developed as one of the strongest and most efficient algorithms in existence today, providing improved security and lower implementation complexity. However, as with other symmetric encryption algorithms, the secret key distribution is regarded as a critical flaw in AES. As an asymmetric cryptosystem, RSA overcomes the problem of secret key distribution. The main disadvantage of RSA is that it has a higher processing cost because to its largekey. As a result, using a hybrid encryption system, in which AES is often used to encrypt large data blocks while RSA is utilized for key management and digital signature applications, is the best approach.

References

- [1]Colombo, M., Asal, R., Hieu, Q. H., El-Moussa, F. A., Sajjad, A., & Dimitrakos, T. (2019, July). Data protection as a service in the multi-cloud environment. In 2019 IEEE 12th International Conference on Cloud Computing (CLOUD) (pp. 81-85). IEEE
- [2] Gadekar, D. P., Sable, N. P., & Raut, A. H. Exploring Data Security Scheme into Cloud Using Encryption Algorithms. International Journal of Recent Technology and Engineering (IJRTE), Published By: Blue Eyes Intelligence Engineering & Sciences Publication, ISSN, 2277-3878
- [3] Sudha, I., & Nedunchelian, R. (2019). A secure data protection technique for healthcare data in the cloud using homomorphic encryption and Jaya-Whale optimization algorithm. International Journal of Modeling, Simulation, and Scientific Computing, 10(06), 1950040
- [4] Sajay, K. R., Babu, S. S., & Vijayalakshmi, Y. (2019). Enhancing the security of cloud data using hybrid encryption algorithm. Journal of Ambient Intelligence and Humanized Computing, 1-10.
- [5] Qiu, H., Noura, H., Qiu, M., Ming, Z., & Memmi, G. (2019). A user-centric data protection method for cloud storage based on invertible DWT. IEEE Transactions on Cloud Computing
- [6] Jakimoski, K. (2016). Security techniques for data protection in cloud computing. International Journal of Grid and Distributed Computing, 9(1), 49-56.-
- [7] Sarmila, K. B., & Manisekaran, S. V. (2019, October). A Study on Security Considerations in IoT Environment and Data Protection Methodologies for Communication in Cloud Computing. In 2019 International Carnahan Conference on Security Technology (ICCST) (pp. 1-6). IEEE.
- [8] CryptDICE: Distributed data protection system for secure cloud data storage and computation AnsarRafique DimitriVan Landuyt EmadHeydari BeniBertLagaisseWouterJoosen
- [9] Shukla, Suraj, Detailed Review of Different Security Techniques for Data Protection in Cloud Computing(April20,2020).
- [10] Data Security in Cloud Computing Using Elliptic Curve Cryptography Imran A. Khanand Rosheen Qazi