
Project Report

Secure File Sharing Using Cloud

Team – 16

Devang Vamja (dsv200000)

Jainish Adesara (jxa200032)

Nisarg Patel (nnp210001)

Meet Chanchad (mxc210021)

1. Introduction

In the current era, Cloud has become an irreplaceable part one's life. From storing a large sized backup data to instantly sharing a file to each other, cloud is being used in our daily lives frequently. Invention of cloud computing has solved numerous problems which were hard to deal with. With all the bright side of cloud computing, there are some problems which need to be addressed as well. One of those problems is security issues. A file being shared on a wireless medium is vulnerable to many threats and put risks on its privacy. This project is a try to solve this problem using encryption and key exchange algorithms. This method can also be useful when the users are putting their data in a third-party cloud storage. It will reduce the risk of data theft and data misuse by any person with mal-intent. This threat keeps on increasing as we increase the size of organization. Many large companies and government bodies are researching ways to deal with this kind of dangers and provide more secure transaction of files through cloud. These researches have provided us with some powerful encryption and key exchange algorithms. Some of the algorithms are said to be uncrackable even using current super computers. AES encryption and Diffie Hellman Key Exchange are one of those algorithms which we will use in our project.

2. System Design

Our system is designed such that it can be differentiated in two parts. The main aim for doing that is easier maintenance on later part and work division between group members. The first part is consisting of a computer application. The task of this application is to become a bridge between the web-hosting application and user's computer device. This application will be used to encrypt and decrypt any given file using various keys. The application will be using a symmetric key encryption-decryption technique. The second part will be a web application. This web-application will act as a user portal. The web-application will register any new user. It will also allow users to download any already registered user's public key. Also, the web-application is hosted on a cloud service which enables it to store a file from a registered user. A registered user can also download encrypted file uploaded from any user.

For better understanding of flow of the process, let us consider a simple case. Let suppose a person X is trying to send a file to person Y using our service and both parties have already registered on web-application. Now, person X can download the public key of person Y. Person X will use our computer application to encrypt the file he wants to send, using his private key and person Y's public key. Then person X will upload the encrypted file on the cloud using our web application. Now person Y can see the file in file list of web application and download it. Once person Y has downloaded the file, he will use the computer application and decrypt the file using his own private key and person X's public key. After the decryption, person Y can see the original content sent by person X. Now, for an instance, person Z is trying to pry in the file sent by person X. Even though person Z has access to public key of person X, he will not be able to decrypt the file without private key of person Y.

3. Algorithms Used

3.1. Diffie Hellman Key Exchange

Diffie Hellman key exchange (DH) is also referred as exponential key exchange. This name is related to the fundamentals of this method. It is a cryptographic method which uses different predefined numbers raised to certain powers as keys. These predefined numbers are never transmitted, rather discussed before the transmission begins. Now we will see how the algorithm is used.

Consider the above scenario, Alice and Bob want to communicate using cloud. They have decided the value of P and G . First, they start by generating the private key for themselves. Then they will generate a public key using P , G and their private key. As we can see, for each time of encryption and decryption one private key is needed. At the time of encryption, sender's private key will be used and at the time of decryption receiver's private key will be used. Though values of P , G and public of both parties is known publicly, there is no way to obtain the secret message (3) by using these numbers. There are certain limitations to it as well. Like the decided value of P and G will decide how many users can exist simultaneously in the environment. As they can not be changed later on and increasing their value will also increase the computational power required, a careful planning is needed. The second limitation is the private-key generator. One can say that it is the most vulnerable part of the whole algorithm. If someone has access to it, they can know the method of generating the private key.

3.2. Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) is one of the most widely used encryption methods. It was established by U.S. National Institute of Standards and Technology (NIST). AES is said to be stronger and more secure than the DES or triple DES. This can be achieved by larger key size which prevents the exhaustive key search attack.

The AES consists of three fixed 128-bit ciphers with key size 128, 192 and 256 bits with 10, 12 and 14 rounds of encryption. These rounds are dependent on key length required. Key size can be very large but the block size maximum value can be up to 256 bits. The AES design is based on SPN networks which is also known as substitution-permutation network.

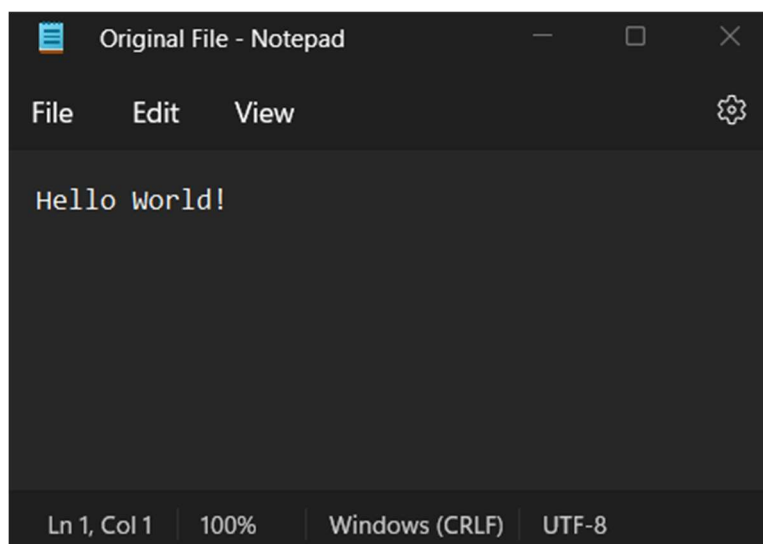
4. Implementation

As we discussed in section 2, our project is made of two parts. The computer application and the web application. To understand the implementation of these parts, let suppose a sender wants to send a file to a receiver. First, sender will visit our web application, where they can see the list of all the available users. From there, sender can download the public key of receiver. Then, sender will use the computer application to encrypt the file using their own private key and receiver's public key. This computer application is build using python modules like tkinter, crypto and secret sharing algorithm. The file will be encrypted using AES. The encrypted file will be then presented to the sender. Now, sender will use the web application to upload the encrypted file on to the cloud.

The web application is constructed using python flask, HTML, CSS and bootstrap. This web application is hosted on a cloud service provider. Receivers can see the list of all user and download the public key of the sender of the file. Then after, receiver can download the encrypted file. Now to decrypt the file, receiver must provide the computer application with their private key and the sender of the file's public key. Then only then, the file will be decrypted successfully and contents can be seen by the receiver. The Web application also have ability to register a new user and provide them with unique private key. Then web application will show the public key of the newly added user in the list of users.

Now let us give an example using some screenshots.

- 1) This is our original file that sender wants to submit.



2) This is where we will encrypt our file with 2 keys.



The image shows a software window titled "Secure file sharing" with standard Windows window controls (minimize, maximize, close). Below the title bar is a "Menu" label. The main content area is divided into two sections:

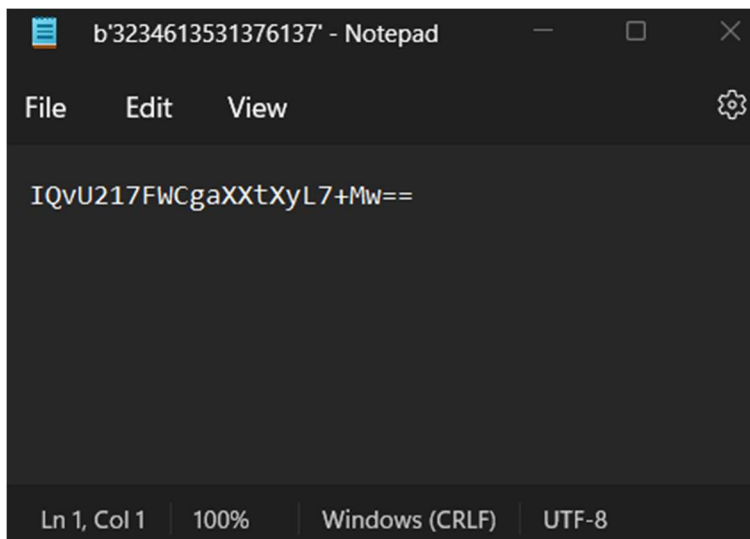
1. File Encryption:

- Select the File: Browse ...
- Save File to: Browse ...
- Public-Key of reciever:
- Private-Key of sender:
-

2. File Decryption:

- Select the File: Browse ...
- Save File to: Browse ...
- Public-Key of sender:
- Private-Key of reciever:
-

3) This is the content of encrypted file.



The image shows a Notepad window titled "b'3234613531376137' - Notepad". The menu bar includes "File", "Edit", "View", and a settings icon. The text area contains the following string:

```
IQvU217FWCgaXXtXyL7+Mw==
```

The status bar at the bottom indicates "Ln 1, Col 1", "100%", "Windows (CRLF)", and "UTF-8".

4) This is the home screen of our web application.

Secure File Sharing

Option 1
File Upload
Upload the encrypted file to the cloud.
[Upload](#)

Option 2
List of uploaded files.
Choose and download the encrypted file from the list of all the files that are present on the cloud.
[Go to Directory](#)

Option 3
Public keys
Download public key of the registered user to use in the encryption of the file.
[Download Public Keys](#)

Option 4
Register user
Register the user to enable access to the system and share the files.
[Register](#)

5) This is Where user can upload the encrypted file.

← → ↻ 📄 localhost:5000/upload-file

🔍 ⚙️ 🌐 📱 🖨️ 🗑️

Secure File Sharing

Please select the file you wish to upload to the cloud. Make sure the file has been encrypted using the standalone application.

[Choose File](#) No file chosen

[Submit](#)

6) The screen after the uploading a file is done.

Secure File Sharing

File Uploaded

Option 1
List Files Get the list of all the different files that are present on the cloud. public-key-directory
Option 2
File Upload Upload the encrypted file to the cloud. Upload-file
Option 3
Go home GO to Home Page Home

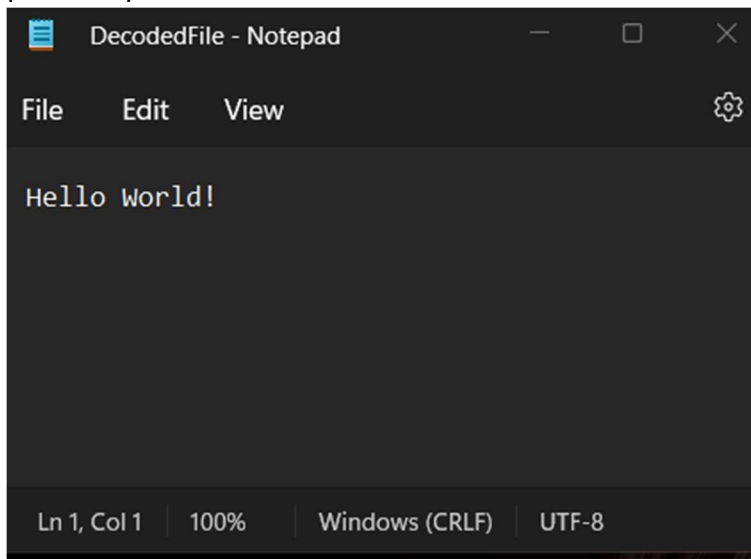
7) List of all the encrypted files which can be downloaded.



8) List of public keys of registered users.



9) Then the download file can be decrypted using the receiver's private key and sender's public key.



10) This is where a user can register.



11) Final screen after registration of new user.

Secure File Sharing

Your one-time generated
For safety purpose, we keep no backup of this key. Refrain yourself from sharing this key.

Your one-time generated private key is:

{{privatekey}}
Note: Copy and save this key.

[Home](#) [close](#)

5. Summary

This system is capable of sending files from a single user to another single user securely using cloud. As AES is highly secure, decrypting a file without proper keys can take more than a million years with highest level of computation power. Though, this environment also presents us with some other challenges as well. This system is not optimal if a user wants to share a file to multiple users. In that case, sender must encrypt the file using all the receivers' public key which will create many redundant data and computational overhead for sender.