CS 6301.005
DEVELOPING AND SECURING OF CLOUD
Dr. Bhavani Turaisingham

Project Report

SECURE FILE STORAGE USING HYBRID CRYPTOGRAPHY ALGORITHMS ON CLOUD

ARJUN HEGDE
SRAVYA KURRA
SURAJ RAGHAVENDRA VADVADGI

ABSTRACT

Cloud computing is the provision of on-demand computing resources - from software to storage and processing capacity - usually over the Internet and on a pay-as-you-go basis. It is now an evolving form of computer science (IT industry's next generation technology). Now a days, it is used for various services and the storing of vast quantities of data in various fields such as business, military, college, industry and so on. Upon request of the users, data stored in this cloud can be accessed or retrieved without direct access to the server computer. But the key concern about online data storage on the cloud is security. This security problem can be addressed in different ways, with cryptography and steganography being the most widely used techniques. But a single technique or algorithm alone may often fail to provide high-level protection. And we have implemented a new security framework that uses a combination of multiple cryptographic algorithms.

In this proposed method, algorithms Fernet(Symmetric Encryption), MultiFernet(same as Fernet with key rotation), Authenticated encryption with associated data (ChaCha20-Poly 1305, AES-GCM, AES-CCM) are used to provide data protection. Key information should show the encrypted portion of the code, the algorithm and the key for the algorithm. During encryption the file is split into N parts. These individual parts of the file will be encrypted simultaneously using different encryption algorithms. Our approach ensures better privacy and security of customer data by storing encrypted data on a single cloud server, using above mentioned algorithm.

Introduction

With the exponential increase in the amount of data being generated by each user every day, it's become impossible to contain the data on one single device, PC or any storage device. With the advent of a promising technology– Cloud Computing, and its benefits like large storage space, low investment cost, virtualization, resource sharing, etc. users can store a vast amount of data and information in the cloud and access it from anywhere, anytime on a pay-per-use basis. Ease with which many users using the cloud for sharing or accessing of information, brings with it, the problem of illegal access and tampering of data in the cloud.

One of the main drawbacks with cloud is security. Technological developments lead to patterns and changes that boost the quality of life. In this fast-paced life where everyone uses a smartphone and has access to the internet, the main concern facing the people is about the protection of their online present information. This security issue also concerns the file which is stored online in a cloud. This security issue also concerns the file which is stored online in a cloud. This can be solved using cryptography.

Techniques of cryptography transform the original data into text in Cipher. Thus, only authorized users with the right key can access data from the server for cloud storage. The cryptography's main objective is to preserve data protection from hackers, online / software crackers and any third-party users. Non-legitimate consumer access to the information leads to confidentiality loss. Security has the features to prevent or avoid this kind of unauthorized access or any other kind of malicious attacks on the data by protecting the trust of the users here.

Privacy is a critical factor of the cloud computing world because of the importance of the information stored on the cloud and the different services offered to users. This data can be highly sensitive and confidential. Therefore, the data protection and security should be fully accurate. The data in the cloud need to be secured against malicious attacks. So, we implemented a new method for data protection, in which we use a combination of multiple cryptography algorithms.

In this proposed method, algorithms Fernet(Symmetric Encryption), MultiFernet(same as Fernet with key rotation), Authenticated encryption with associated data (ChaCha20-Poly 1305, AES-GCM, AES-CCM) are used to provide data protection. Key information should show the encrypted portion of the code, the algorithm and the key for the algorithm. During encryption the file is split into N parts. These individual parts of the file will be encrypted simultaneously using different encryption algorithms.

Cryptographic Algorithms

cryptography includes both high level recipes and low level interfaces to common cryptographic algorithms such as symmetric ciphers, message digests, and key derivation functions.

Below are the cryptographic algorithms used in this project:

❖ Fernet MultiFernet:

Fernet is an implementation of symmetric (also known as "secret key") authenticated cryptography. It guarantees that a message encrypted using it cannot be manipulated or read without the key. To use this algorithm, we used methods like generate_key(), encrypt and decrypt methods

Pseudo code:

```
key = Fernet.generate_key()
f = Fernet(key)
token = f.encrypt(text)
f.decrypt(token)
```

❖ MultiFernet

MultiFernet is same as Fernet except that it rotates the key. It performs all encryption options using the first key in the list provided. MultiFernet attempts to decrypt tokens with each key in turn. Key rotation makes it easy to replace old keys. One can add your new key at the front of the list to start encrypting new messages, and remove old keys as they are no longer needed.

Pseudo code:

```
key1 = Fernet(Fernet.generate_key())
key2 = Fernet(Fernet.generate_key())
f = MultiFernet([key1, key2])
token = f.encrypt(text)
f.decrypt(token)
```

❖ Authenticated Encryption with Associated Data

Authenticated encryption with associated data (AEAD) are encryption schemes which provide both confidentiality and integrity for their ciphertext. They also support providing integrity for associated data which is not encrypted.

Below are the AEAD algorithms that we have used in this project

- ChaCha20-Poly1305:

The ChaCha20Poly1305 construction is defined in RFC 7539 section 2.8. It is a stream cipher combined with a MAC that offers strong integrity guarantees.

Pseudo code:

```
chacha = ChaCha20Poly1305(key)
secret_data = chacha.encrypt(nonce, raw, aad)
```

- AES-GCM

  The AES-GCM construction is composed of the AES block cipher utilizing Galois Counter Mode (GCM).

  Pseudo Code:

  ```
  aesgcm = AESGCM(key)
  secret_data = aesgcm.encrypt(nonce, raw, aad)
  ```

- AES-CCM

  The AES-GCM construction is composed of the AES block cipher utilizing the AES-CCM construction is composed of the AES block cipher utilizing Counter with CBC-MAC (CCM) (specified in RFC 3610).

  Pseudo Code:

  ```
  aesccm = AESCCM(key)
  secret_data = aesccm.encrypt(nonce, raw, aad)
  ```

  For decryption, the reverse process of encryption is used

Implementation

This section describes the methodologies and implementation details of the targeted framework. The contents in this section are organized as follows: Section 6.1 contains the detailed methodology of targeted framework. Section 6.2 describes the details of how to run our project files. Section 6.3 enlists the python files and its file. Section 6.4 shows the overall working of the project using web GUI.

Methodology

- Load the file on the server.
- Divide the uploaded file into N parts.
- Encrypting all the parts of the file using any one of the selected algorithms (Algorithm is changed with every part in round robin fashion).
- The keys for cryptography algorithms is then secured using a different algorithm and the key for this algorithm is provided to the user as public key.
- After the above 4 steps you will have a N files which are in encrypted form which are stored on the server and a key which is downloaded as public key for decrypting the file and downloading it.
- To restore the file, Load the key on the server.
- Decrypt the keys of the algorithms.
- Decrypt all the N parts of the file using the same algorithms which were used to encrypt them.
- Combine all the N parts to form the original file and provide it to the user for downloading.

Execution:

The project is based on Python 2.7.15 platform running it on any other platform might create some issues.

Step 1: Install Requirements

pip install -r requirements.txt

Step 2: Run the application

python application.py

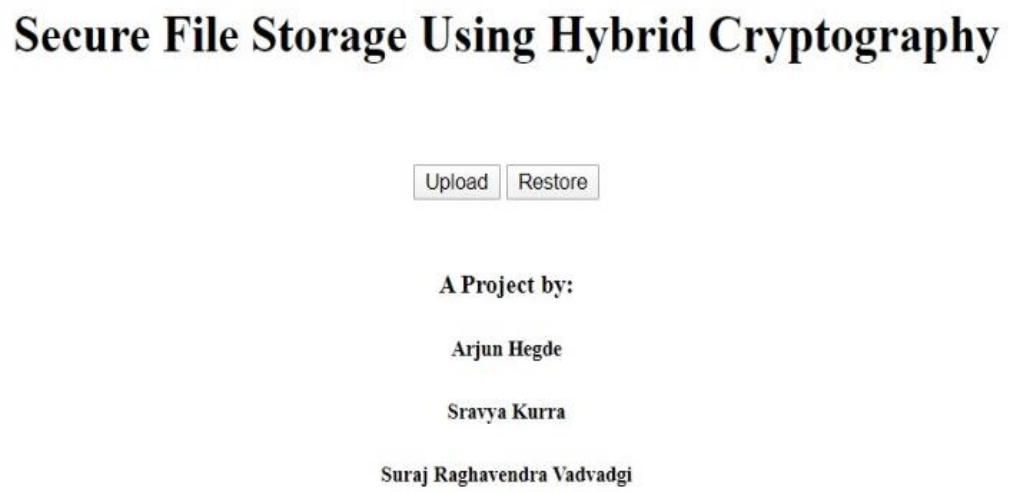
Step 3: Visit the localhost from your browser

Back-end Application

- application.py (main function that calls all required functions)
- encryption.py (encryption algorithms definitions)
- decryption.py (decryption algorithm definitions)
- divider.py(divides the file into parts)
- restore.py(restoring the file for download once key is submitted)
- tools.py(file parts management and storage in directory)

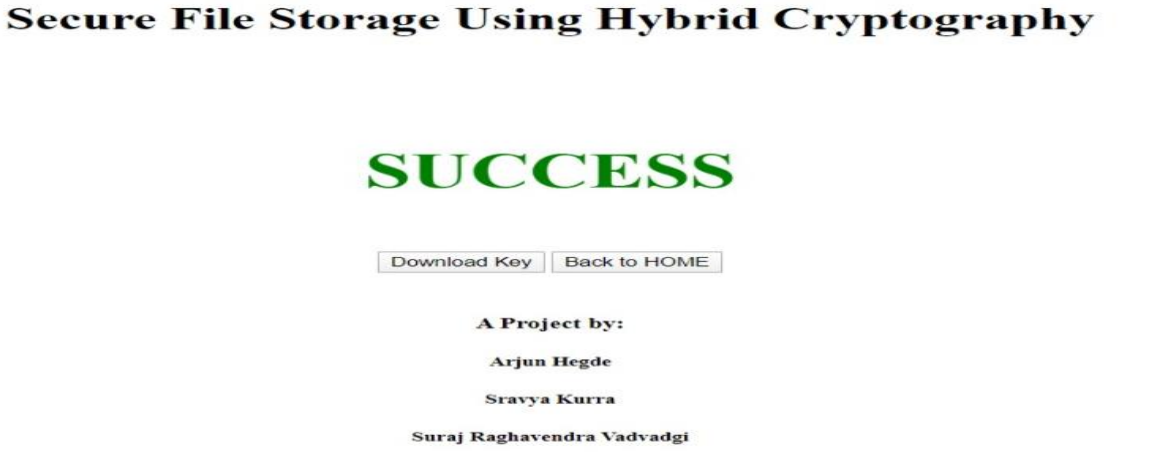
Web-application

The web application of the project was used for secure storage on the cloud. The major steps included in the web application are as follows:

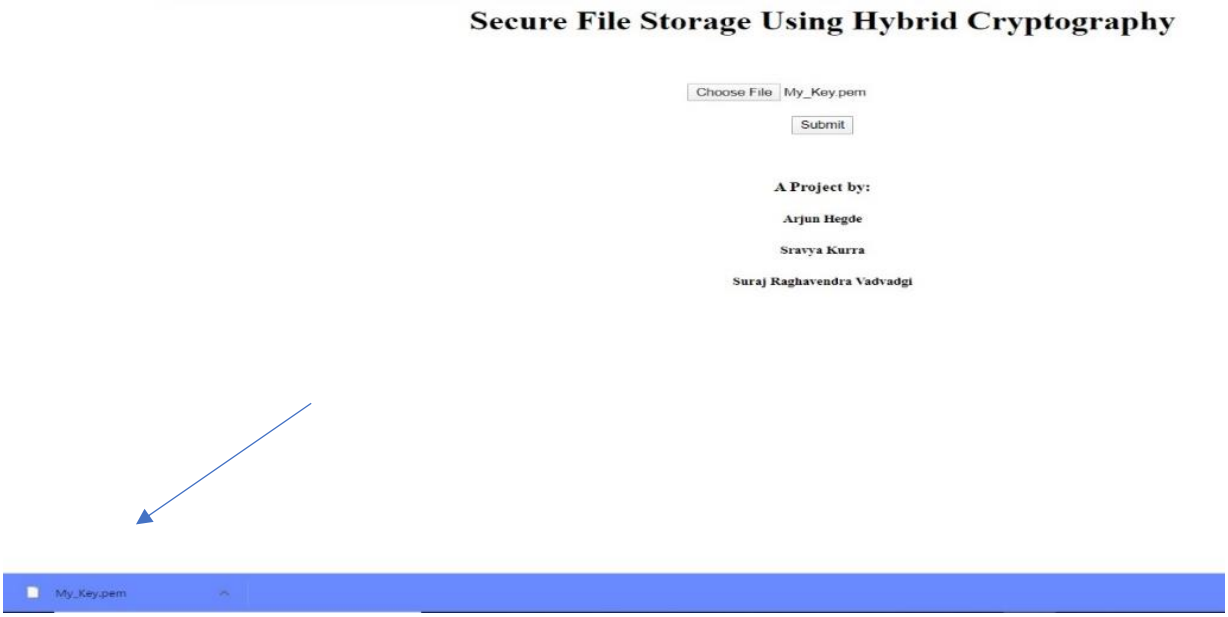


- GUI for the user to upload the file for secure storage.

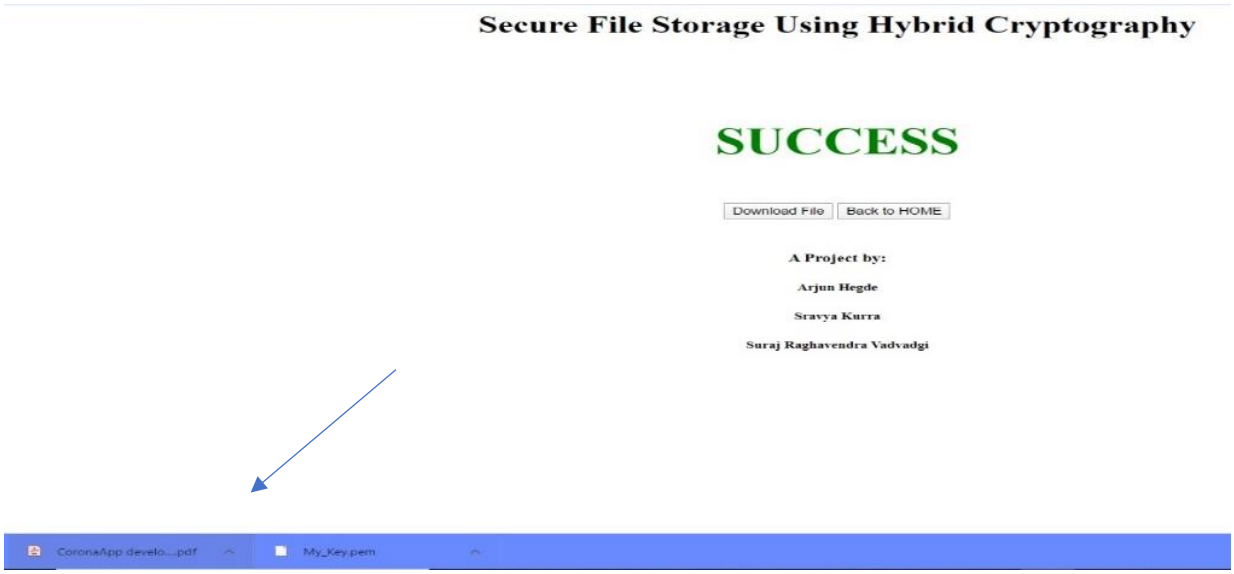
- The user will choose a file from the system and press submit to securely store his file on the cloud.



- Once the user uploads the file successfully, he/she gets the option to download public key which will be used to download the file later when needed.



- Upload the downloaded key to restore/get the original file from cloud storage.



- Once the user submits the public key, he/she will get option to download the file.

7.0 Conclusion

The main aim of this program is to safely store and retrieve data in the cloud which is managed only by the data owner. Data security problems in the cloud storage are addressed in n number of ways. Data protection is accomplished using cryptography and steganography techniques. Use of a single algorithm is not effective for high level security to data in cloud computing. The proposed model efficiently secures the file/data on cloud. We've achieved better data integrity, high reliability, low latency, authentication and confidentiality with the aid of the proposed protection framework. We should introduce public key cryptography in the future to prevent any attacks during the transmission of the client data to the server.

References

1. Using Cryptography Algorithms to Secure Cloud Computing Data and Services
   Eng. Hashem H. Ramadan, Moussa Adamou Djamilou
2. Security in Cloud Computing using Cryptographic Algorithms
   Shakeeba S. Khan1 , Prof.R.R. Tuteja2
3. https://cryptography.io/en/latest/hazmat/primitives/aead/
4. https://cryptography.io/en/latest/fernet/