

1. INTRODUCTION

1.1 What is Cloud ?

Cloud computing has recently surfaced as a new paradigm for hosting and delivering services over the web. Cloud computing represents to both the applications delivered as facilities over the web and the hardware and systems software in the data centres that deliver those services.

1.1.1 Cloud Definition

Cloud computing is shared pool of configurable automatic data processing higher-level services and system resources that can be quickly and rapidly delivered with negligible management effort, usually over the web. Cloud computing depends on sharing of resources to realize coherence and economies of scale, just like a utility.

NIST definition of cloud computing *Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., storage, servers, networks, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*

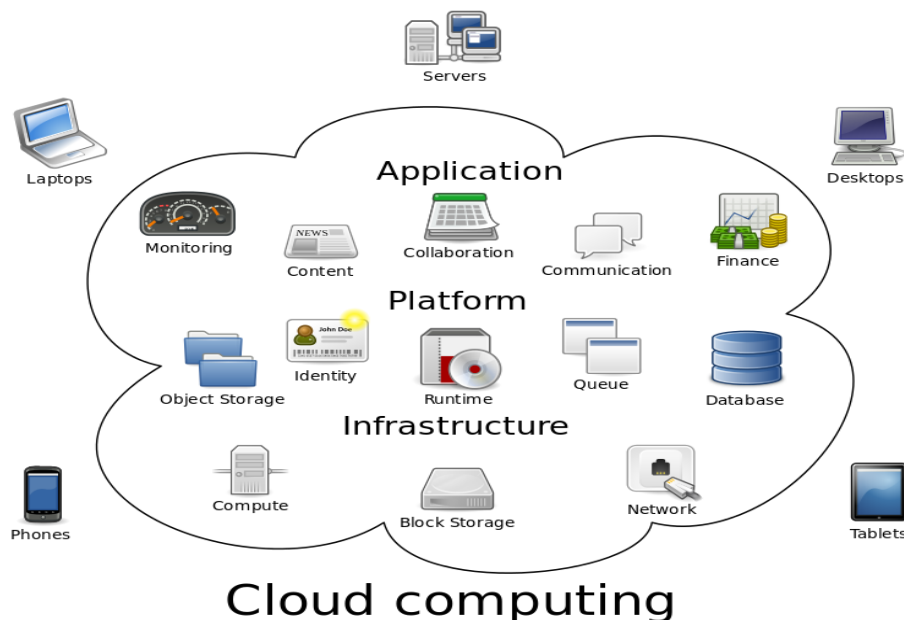


Figure 1.1 Overall view of Cloud Computing

1.1.2 Cloud Architecture

Cloud generally provides three services, namely, *Software as a Service (SaaS)*, *Platform as a Service (PaaS)* and *Infrastructure as a Service (IaaS)*.

- *Software as a Service (SaaS)*: SaaS permits people to use cloud-based web applications. E-mail services such as Hotmail and Gmail are examples of Software as a Service (SaaS).
- *Platform as a Service (PaaS)*: PaaS denotes cloud platforms that provide runtime environments for developing, testing and managing applications, examples of *PaaS* would include Heroku and Google App Engine.
- *Infrastructure as a Service (IaaS)*: IaaS is a cloud facility and service that offers basic computing storage, infrastructure, servers, and networking resources. To paraphrase this, IaaS is a virtual data centre wherein major IaaS providers include Amazon Web Services, Microsoft Azure, and Google Compute Engine.

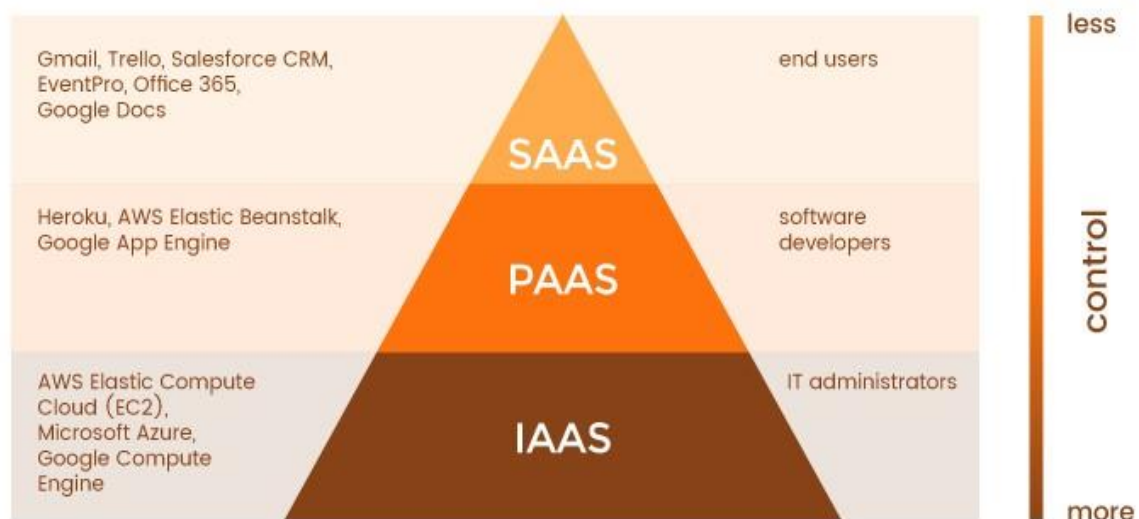


Figure 1.2 Cloud Architecture

1.1.3 Characteristics of Cloud Computing

Cloud computing has become a popular and successful business model due to its attractive features.

- On-demand self-service

- Pooling of Resources
- Provision of Broad network access
- Rapid elasticity
- Measured service

1.2 Need of Security in Cloud

Cloud computing has become a popular and successful business model due to its attractive characteristics and features. In addition to the reimbursements, the features mentioned above also result in severe specific to cloud security issues. The individuals whose worry is the cloud security still hesitate to transfer their business to cloud. Security issues have been the governed barricade of the growth and worldwide use of cloud computing, namely:

- **Outsourcing:** Outsourcing in cloud refers to providing data to a third-party cloud hosting provider to support and deliver IT services that could be handed over inhouse. Outsourcing means that the clients physically lose control on their tasks and data. The problem concerning loss of control has become one of the main and root causes of cloud insecurity.
- **Multi-tenancy:** Multi-tenancy implies that the cloud platform is distributed and shared and exploited by multiple clients. Furthermore, in an environment that is virtual, data related to different clients may be located on the same physical machine by assertive resource allocation policy. A series of security problems and issues such as computation breach, data breach, flooding attack, etc., are experienced.
- **Massive data and intense computation:** Cloud computing can manage intense computing tasks and mass data storage. Therefore, old-style security mechanisms may not avail due to unbearable communication or computation overhead. For instance, to verify the veracity of data that is stored remotely, it is not practical to hash the entire data and information set. For this reason, new approaches and protocols are expected.

Security of system can likewise defend security of information in transit, procedures and capacity. Security notions of access, approval and authorization control are required to be executed to enhance the security of the information that is extracted on cloud.

The three key zones in Data Security are shown in figure 1.3.



Figure 1.3 Areas in Data Security

1.3 Cryptography

Converting useful data into unreadable text so that no one else can read it except the pre-determined user is encryption and techniques used are called cryptography techniques. It may be accomplished through scrambling words, using code words or using highly efficient mathematical techniques. There are different algorithms classified in two-

- a. Symmetric Algorithms – This algorithm uses same key to encrypt and decrypt the message. For instance, if some user wants to send a message to another and wants nobody else should read it. So, he can encrypt the data using a secret key that can be shared with the receiver who will decipher the data using the same key. As can be judged the key has been shared with all the users who ought to retrieve data.
- b. Asymmetric Algorithms – These types of algorithms make use of two separate keys in process of decryption and encryption respectively. Like a user ciphers the data using one key (known as public key) and receiver decrypts the message using separate key (known as private key).

- i. Public Key – This key is accessible to all over the internet and used to cipher the secret text.
- ii. Private Key – As clear from the name it is a receiver's private key and only the receiver will be able use this key to decrypt the required message.

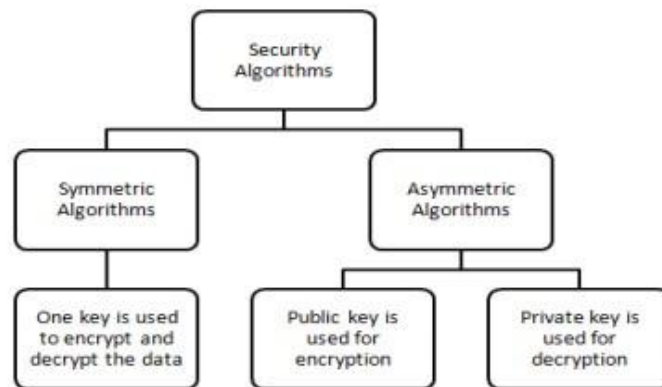


Figure 1.4 Security Algorithms

1.3.1 Types of Cryptography

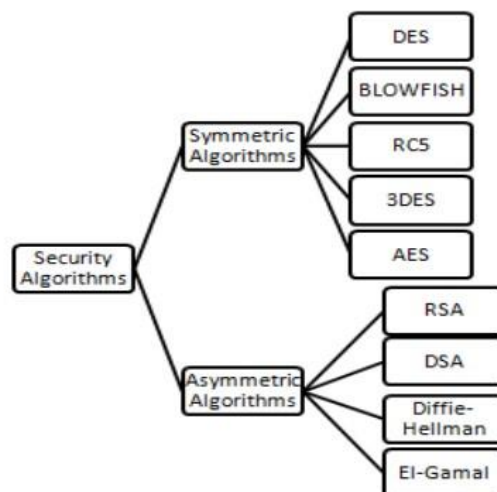


Figure 1.5 Classification of Cryptographic Algorithms

1.3.2 Hybrid Cryptography

Hybrid cryptography refers to the use of two or more encryption techniques, thus, making cloud robust and securing the data or privacy.

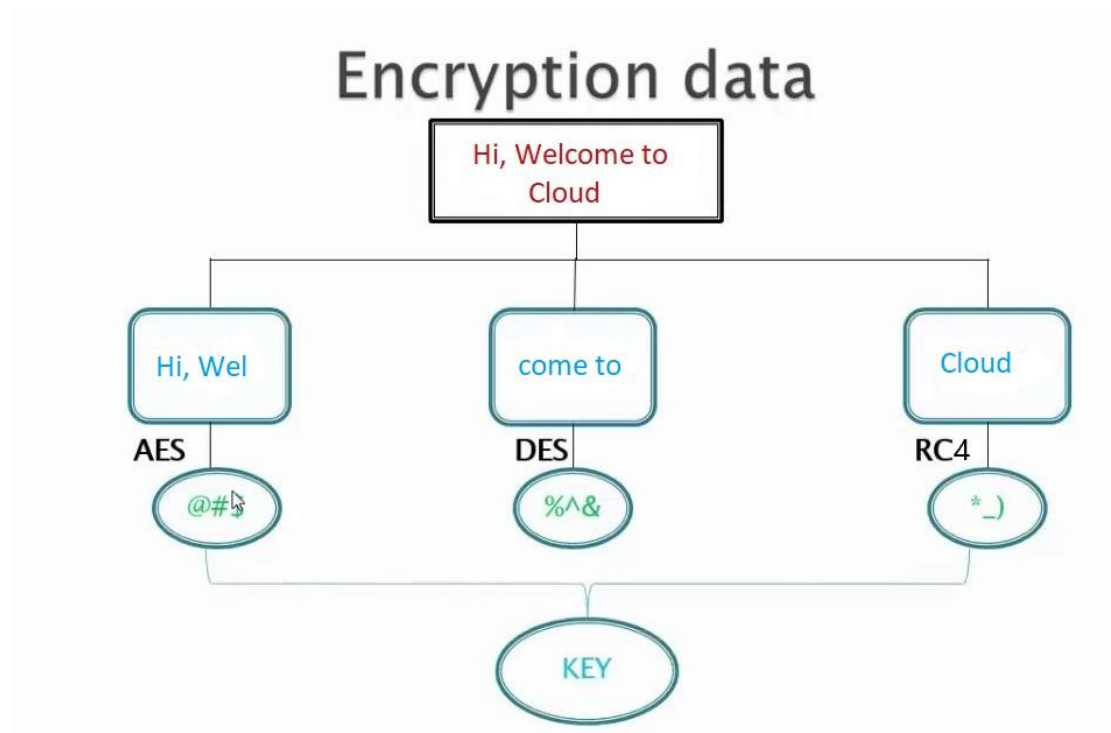


Figure 1.6 Encryption of data using Hybrid Technique

1.3.3 Need of Hybrid Cryptography

We know that classic encryption schemes have long been used for security purposes and they have been successful to some extent. So why use hybrid system. The basic answer is security. Basically, what hybrid cryptography does is that it mixes the effectiveness of symmetric encryption with easiness of public key encryption. It enhances security level and both types of encryption efficiency. It has many advantages. Firstly, it enables user to communicate through hybrid cryptography. Some may think asymmetric scheme will hinder speed of encryption but that is not the case. Simultaneously using symmetric scheme, it increases efficiency of system as well as both schemes. It provides better security and increased performance.

1.4 Problem Statement

The huge chunks of data that needed to be processed and stored were impossible for storage in physical drives on computers which helped paved the way for Cloud technology. The new technology was fast, dependable, and data sharing was efficient. There were certainly many benefits to it and also had its limitations. Limitations were harmful to data owner. As the data

was stored on outside system not within the physical reach of user there are the concerns for security of data. Data sharing may also lead to breach of sensitive data and privacy of user. There have cryptographic techniques to cipher the data and code it so that the attacker cannot understand the data.

Techniques like DES, its more efficient 3DES, AES, RSA, RC4 and many others have proven to be a success in hiding the data and securing it. Everything is prone to attack and so are they. Every individual encryption technique is known to be attack prone due to increase in computational power days. Any new technique which will be developed will be known to attacks but to increase security and efficiency hybrid encryption is used. Hybrid encryption provides efficiency of public key cryptography and easiness of private key cryptography. Here in this project we incorporated AES, DES and RC4 techniques to implement hybrid cryptography.

2. Technology & Literature Survey

HTML

- HTML stands for Hyper Text Markup Language. It is used to design web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. Markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most of markup (e.g. HTML) languages are human readable. Language uses tags to define what manipulation has to be done on the text. HTML is a markup language which is used by the browser to manipulate text, images and other content to display it in required format.

CSS

- Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.
- CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.
- CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.
- You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want. To make a global change, simply change the style, and all elements in all the web pages will be updated automatically. CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes. Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

With ever increasing size of data in turn of century ‘Cloud Computing’ became the most essential resource in resource sharing and data storage applications. Obviously, data security concerns were raised. Given literature review of research papers of various authors discuss motive of Cloud Computing and emphasizes its need. In the implementation and following analysis data security has been the forefront issue and hence to overcome this we give various cryptographic algorithms individually and then permute the best combination among those to achieve security on cloud.

2.1 TITLE: “Security and Privacy in Cloud Computing” [2]

This research paper introduces cloud architecture and characteristics of cloud. Through the paper authors identify present security and privacy issues and discuss their proneness to attack and provide currently used defense mechanisms and analyze their efficiency in doing the required work.

2.1.1 Cloud Characteristics

- a) **On-demand Self Service:** A cloud user can get the necessary data, software, server access from anywhere anytime without concerning with cloud provider.
- b) **Broad Network Access:** As all the liable services and data is available through Internet user may access it using versatile devices like computers, phones, PDA's.
- c) **Resource Pooling:** The cloud provider uses multi-user model where it collects all the resources at a server and different customers make use of pooled resources like network bandwidth, storage memory, software service.
- d) **Rapid Elasticity:** Cloud services can anytime be levelled up (i.e. increase services and quality) or levelled down all dependent on whether customer wants to use that resource.
- e) **Measured Service:** Services being provided and those used by the user and owner is monitored, controlled and measured with respect to resource usage.

2.1.2 Security Challenges

The various security challenges have been discussed in the following figure.

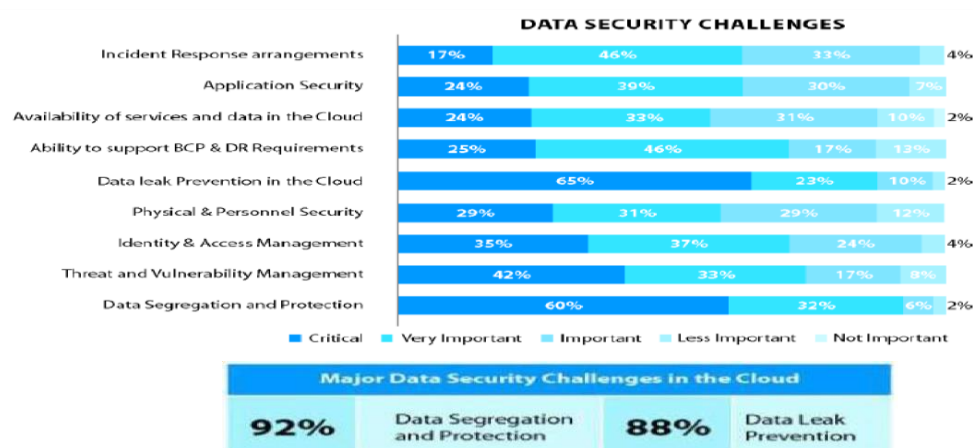


Fig 2.1 Challenges in Data Security on Cloud

As can be seen from above figure issues like outsourcing, multi-tenancy leaves cloud prone to attacks like data breach, computation breach and overload problem.

2.1.3 Current Strategies

Current strategies include Virtualization, Data Center Techniques and Map reduce center which to certain extent helps the cloud vendor in overcoming those challenges, but each have its shortcomings and not useful. Author gives a cloud security ecosystem to model different attacks and use different mechanisms against them.

2.1.4 Conclusion

The paper provided the security challenges that are a big hindrance in the customer trusting the cloud technology for commercial and personal operations. The issues regarding current defense strategies have not been ratified and left open for future resolve of these issues and enhance user trust on cloud.

2.2 TITLE: “Secure Data Sharing in Clouds” [1]

Preserving in thoughts the challenges in cloud the researcher designs a new sedasc methodology for secure facts storage and transfer. The author analyzed like cl-prescheme, certificates much less encryption, el-ghamal cryptography schemes with their blessings and cons. Like the cl-pre-scheme generates a public-non-public key pair and uses bilinear method for encryption which growth cost of encryption. The certificates much less encryption even though tries to enhance upon the cost issue however falls short in trusted facts garage. On the other hand, the el-ghamal scheme uses bilinear and incremental encryption however the complexities nonetheless exist. The studies writer’s scheme provided fine effects and improve upon shortcomings of analyzed schemes.

2.2.1 SeDaSC

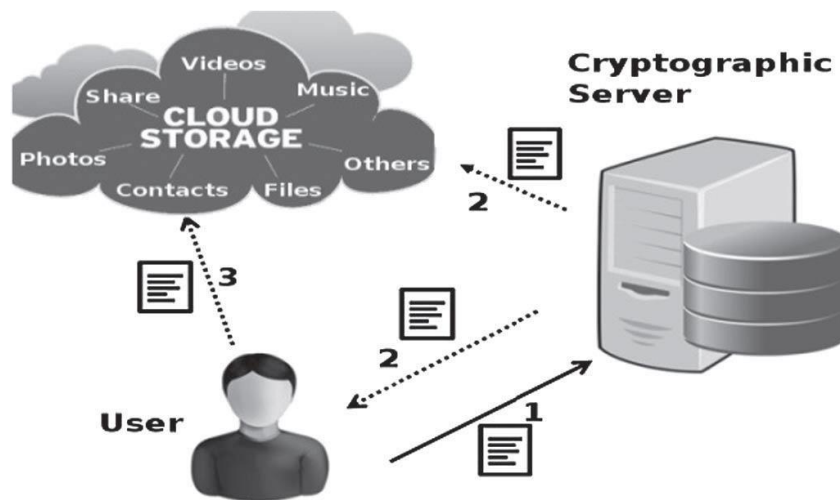


Fig 2.2 Basic Model of SeDaSC

As the figure suggests that,

- i. Users file is encrypted as normal.
- ii. A key is generated which is broken in two shares. iii. One share is with user and the other is saved in cryptographic server.
- iv. The key with user is encrypted with public key of user.

As shown in under discern the comparison in encryption time with key calculation which needs to be performed on every occasion while doing encryption and decryption. The record sizes taken are inside the variety from 0.1 mb to 500 mb. The time varies with document length.

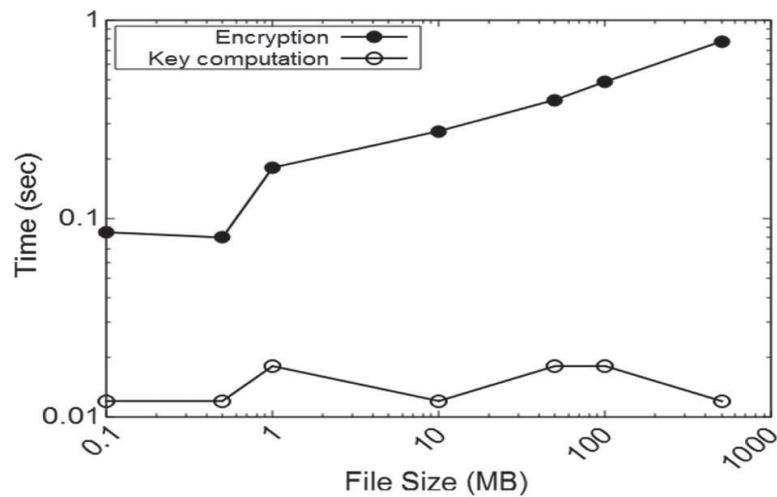


Fig 2.3 Encryption time with key compute.

2.2.2 Conclusion

Although, it is clear from the picture that key computation time changes merely with increase in file size and the encryption time too is commendable with this methodology. But the time needs to check with file sizes of greater than 1GB to really test. Additionally the important element is that it stores key on outside server can be underneath outside threats, so measures should be taken to make it extra comfy or limit the level of trust in the server.

2.3 TITLE “A Study of Encryption Algorithms (RC4, DES, 3DES and AES) for Information Security” [7]

This paper gives a comparative view of different encryption algorithms on parameters like CPU usage, encryption time, ROM utilization, throughput with different file sizes, length of packet and data type.

2.3.1 Rivest-Shamir-Adleman (RSA)

It is a public cryptosystem technique that incorporates block size encryption and variable key size. The steps involved are-

- i. Generate two distinct prime numbers.

- ii. Calculate t as product of two. iii. Now compute $\phi(t)$. iv.
- Find d such that $d \cdot e = 1$
- v. Public Key is $(1, e)$ and Private key is $(1, d)$.

Its most apparent disadvantage is if two numbers are of massive length then it takes more time and, they should be of comparable size.

2.3.2 Data Encryption Standard (DES)

This encrypt algorithm is the most widely used because it works on bits. It is a block cipher incorporating feistel structure. Length of message at one time to be encrypted is 64 bits and the key size of 64-bits but every 7x bit is a check bit which is removed in making of sub-keys. The steps involved are-

- i. Cut the plain text in two parts left and right.
- ii. Design 16 sub keys 48 bits long each. iii. Code words are written for each 64-bit block of data.

As known, it is one of the best algorithms because no such possible attack is known to crack it other than brute-force which is costly and time consuming.

2.3.3 Triple DES

Alias 3 DES, it is extension of DES because it has greater key length. DES was more prone to brute attacks due to increasing computational power. The greater key length provides more security as it uses 3 keys for data security i.e. 168 bits long key size means more permutations and harder for attackers.

The steps are-

- i. Encrypt using first key.
- ii. Now decrypt using second for compatibility with previous.
- iii. Again, encrypt with the help of third key.
- iv. $\text{code} = \text{Ecr}_3(\text{Dcr}_3(\text{Ecr}_1(\text{text})))$

Although it is better version of standard DES, but it can be breached using MITM attacks. So, to defend better against them it can be implemented using secret key size of 112-bits.

2.3.4 Advanced Encryption Standard (AES)

AES also known as Rijndael structure is an iterative algorithm rather being called a fiestel structure. It also a block cipher has block capacity of 128 bits and key size any of 128, 192 or 256 bits. It is iterative because it uses rounds to convert data to cipher text depending on key sizes.

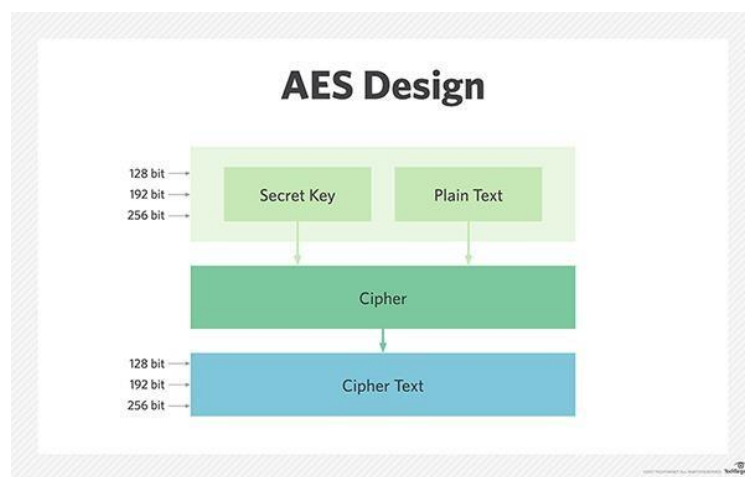


Figure 2.4 AES Design

2.3.5 RC4

RC4 is produced by Bokkos Rivest also called Rivest Cipher four. Here the stream figure is utilized for secret composing of the plain content. Pseudorandom stream of bits' square measure produced by the RC4 algorithm, and bit-wise encryption or decryption has been achieved and performed. The generation key framework includes 2 phases; one is that the stage of each of the 256 bytes Another is 2 8-bit file pointers. The key length for the given RC4 is between 40-128 bits. On the off chance that the basic square figures don't appear to utilize mackintosh effectively, bit-fluttering assault is doable and furthermore the stream figure assault is moreover powerless on the off chance that they're not legitimately authorized.

2.3.6 Blowfish

Also, a symmetric cipher built as an alternate for more commercial AES and DES with varying key sizes of length from 32 to 448 bits with a block size of 64 bits. It's a 16round cipher. The steps involved are-

- i. left half of data is xored with right p^{th} array.
- ii. This data is given as input to blowfish's function.
- iii. Output of this function is xored with right half of data.
- iv. Left and right are swapped.

It is efficient cipher and can be used commercially as alternative to AES. Most possible obvious attacks on this encryption are birthday attacks and should not be used for files of size more than 4gb.

2.3.7 Conclusion

After studying different encryption schemes, we get to know the pros and cons of each scheme and possible attacks on each type. The schemes described have different encryption times and behave differently with different material resources like file size, packet size and processor speed.

3. SYSTEM DESIGN

3.1 Software Requirements

- PyCharm Environment
- Chrome or Microsoft Edge Browser

3.2 Hardware Requirements

- Random Access Memory (RAM): 1 GB or above
- Central Processing Unit (CPU): 1.7 GHz Processor and above
- Operating System (OS): Windows 8 and above

3.3 Proposed Model

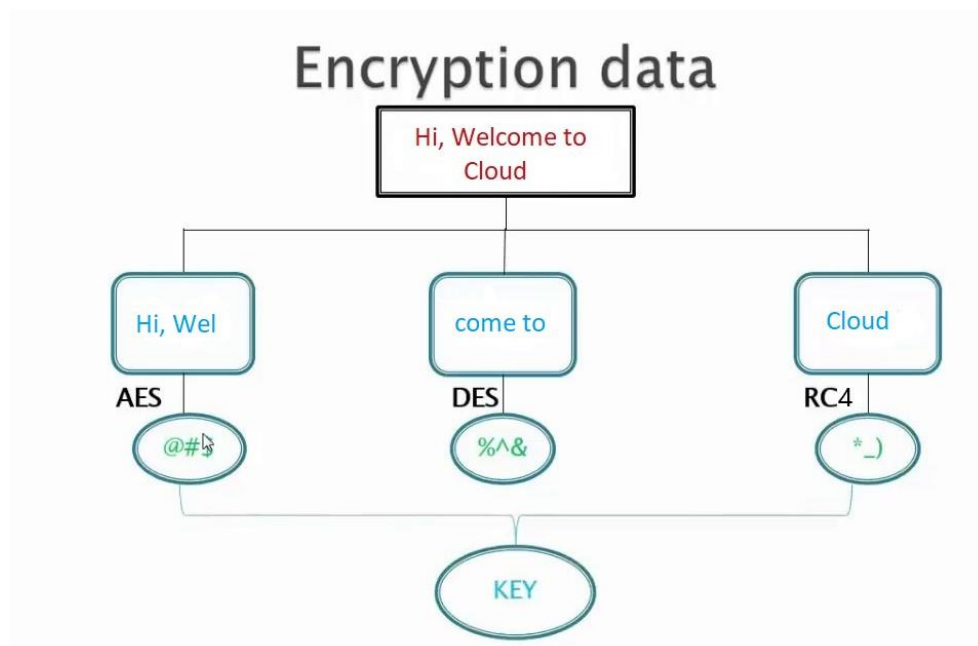


Figure 3.1 Hybrid Encryption

3.4 Flow Diagram

The given file for encryption is encrypted using the hybrid encryption as shown in figure 3.1 above. The file is selected and diving into three equal parts using the file system module. Then, each part is encrypted using the AES, DES and RC4 encryption techniques. The encrypted

parts are then merged and saved into a single file which, then, can be uploaded on the cloud servers.

Encryption of the file is carried out as shown:

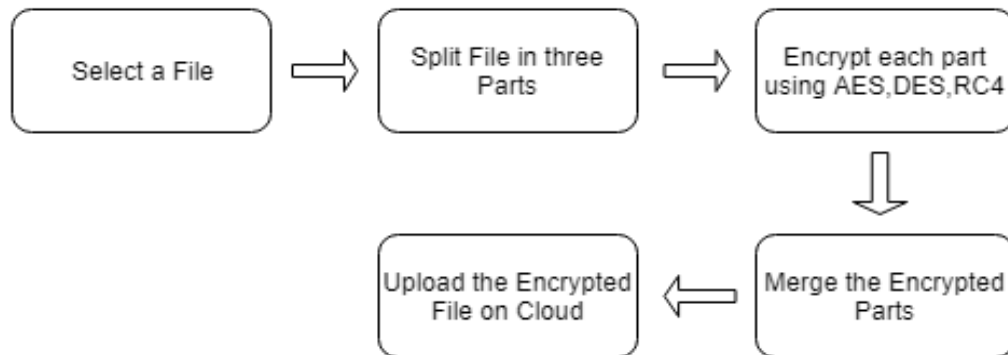


Figure 3.2 Encryption Flow Diagram

For, decryption the encrypted file is download from the cloud servers and then split into three parts using a certain special character into three parts whereupon each part is then decrypted using the same techniques which were used for encryption, i.e. AES, DES and RC4. The decrypted parts are merged into one and the retrieved file can then be used.

Decryption of the file is carried out as shown:

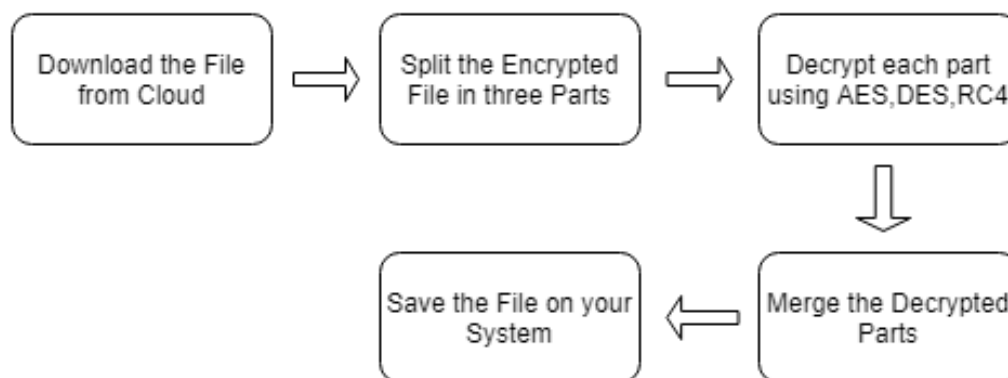


Figure 3.3 Decryption Flow Diagram

3.5 Research Algorithm

3.5.1 AES (Advanced Encryption Standard)

AES the fiestel structure is iterative scheme of cipher conversion. AES is also acknowledged as Rijndael. AES was founded and established by the U.S. National Institute of Standards and Technology (NIST) in 2001. It is a symmetric block cipher.

AES Cipher Details

Key Sizes 128, 192 or 256 bits

Block Size 128 bits

Structure Substitution-permutation Network

Rounds 10, 12 or 14 (depending on key-sizes)

The AES structure is described as-

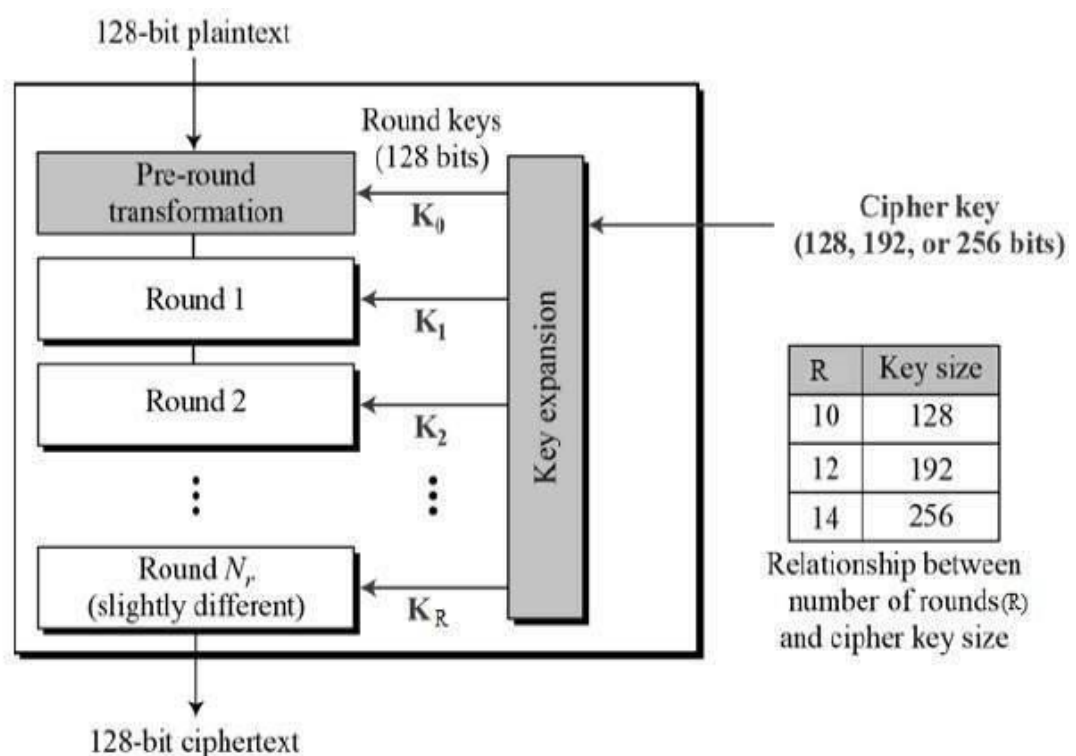


Figure 3.4 Basic Flow AES

Each Basic Process is as shown –

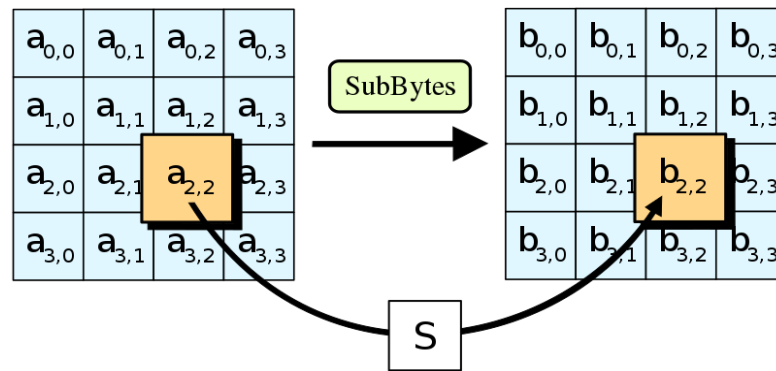


Figure 3.5 The SubBytes Step

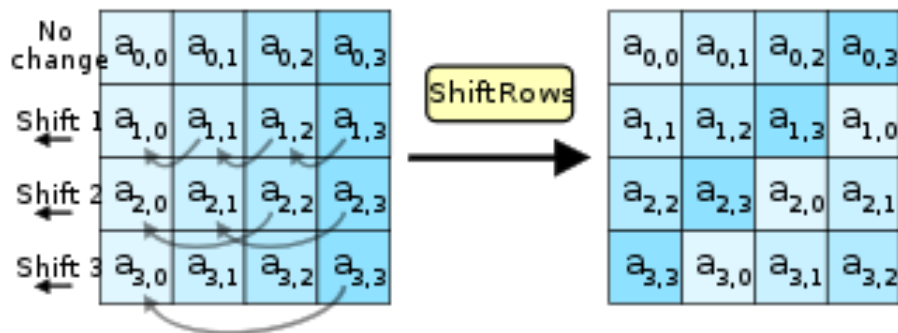


Figure 3.6 The ShiftRows Step

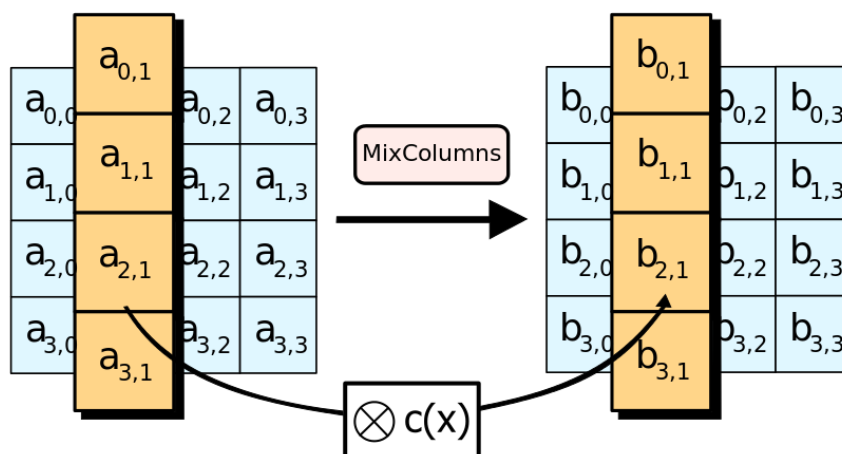


Figure 3.7 The MixColumns Step

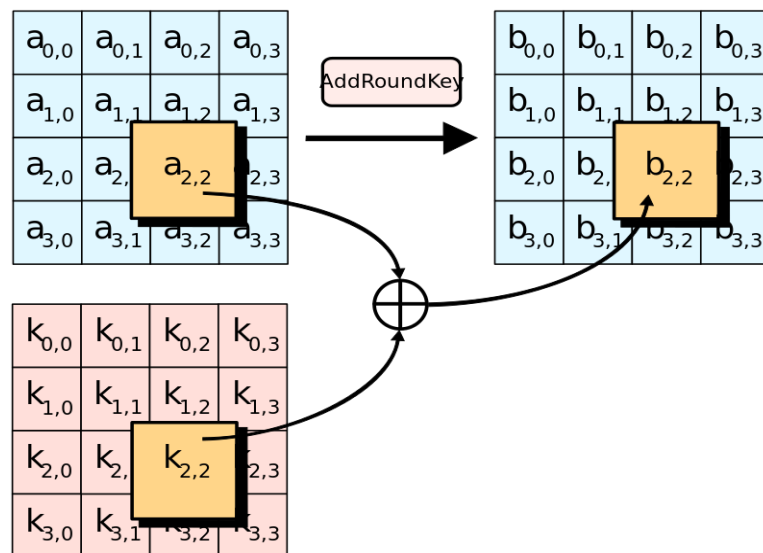


Figure 3.8 AddRoundKey Steps

3.5.2 DES (Data Encryption Standard)

It is a block cipher incorporating feistel structure. Length of message at one time to be encrypted is 64 bits and the key size of 64-bits but every 7x bit is a check bit which is removed in making of sub-keys. Every step involved is shown in flow graph below.

Like the 64-bit data that is to be ciphered is passed through init permutation box which is inverse if final box. It has 16 rounds

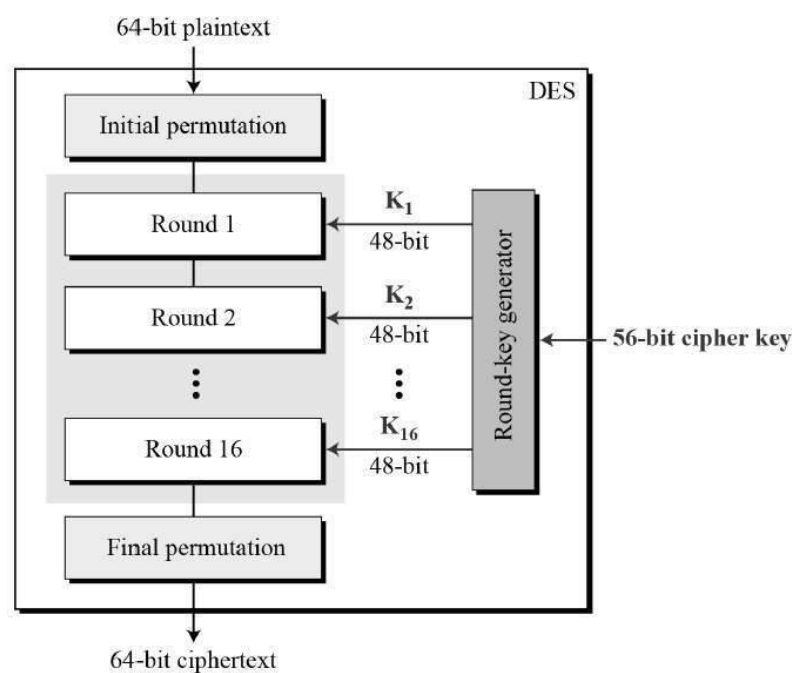


Figure 3.9 DES Flow Diagram

DES Cipher Details

Key Sizes 56 bits

Block Size 64 bits

Structure Feistel structure

Rounds 16

3.5.3 RC4 (Rivest Cipher 4)

RC4 is a public key cryptography. It is a streamed cipher that is key stream which is basically the size of 64 to 128 bytes for each byte of text is generated. This key stream bit by bit is xored with the plain text to form the encrypted data. The key generated is formed in two steps know as key scheduling algorithm and pseudo random number generation and is between 40 -2048 bits. It is easy to implement algorithm yet is a very strong cryptographic technique. It is used WEP for 802.11 networks in SSL. If the common block ciphers don't seem to be used mackintosh powerfully, bit-flapping attack is feasible, and the stream cipher attack is additionally vulnerable if they're not properly enforced.

Schematic Representation of RC4

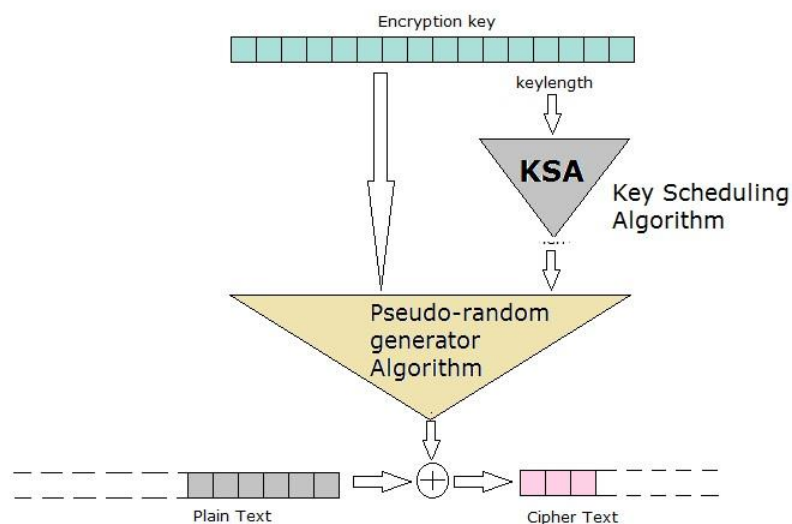


Figure 3.10 RC4 Schematic Representation

RC4 Cipher Details

Key Sizes 40–2048 bits

Stage Size 2064 bits (1684 effective)

Type Type Shared Key Stream, Symmetric

Rounds 1

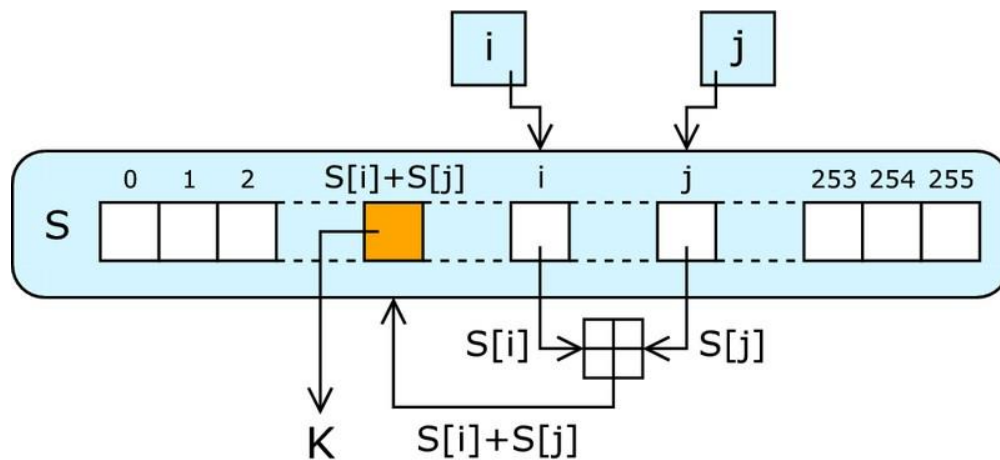


Figure 3.11 RC4 Lookup Stage

3.5.4 RSA (Rivest–Shamir–Adleman)

It is one of the first public cryptosystem technique that incorporates block size encryption and variable key size. In this kind of encryption, the encryption key is public and varies from the decryption key which is kept a secret. The process's asymmetry is based on the practical difficulty of the factorization of the product of two large prime numbers. The steps involved are-

- i. Generate two distinct prime numbers.
- ii. Calculate t as product of two.
- iii. Now compute $\phi(t)$.
- iv. Find d such that $d \cdot e = 1$
- v. Public Key is $(1, e)$ and Private key is $(1, d)$.

Its most apparent disadvantage is if two numbers are of massive length then it takes more time and, they should be of comparable size.

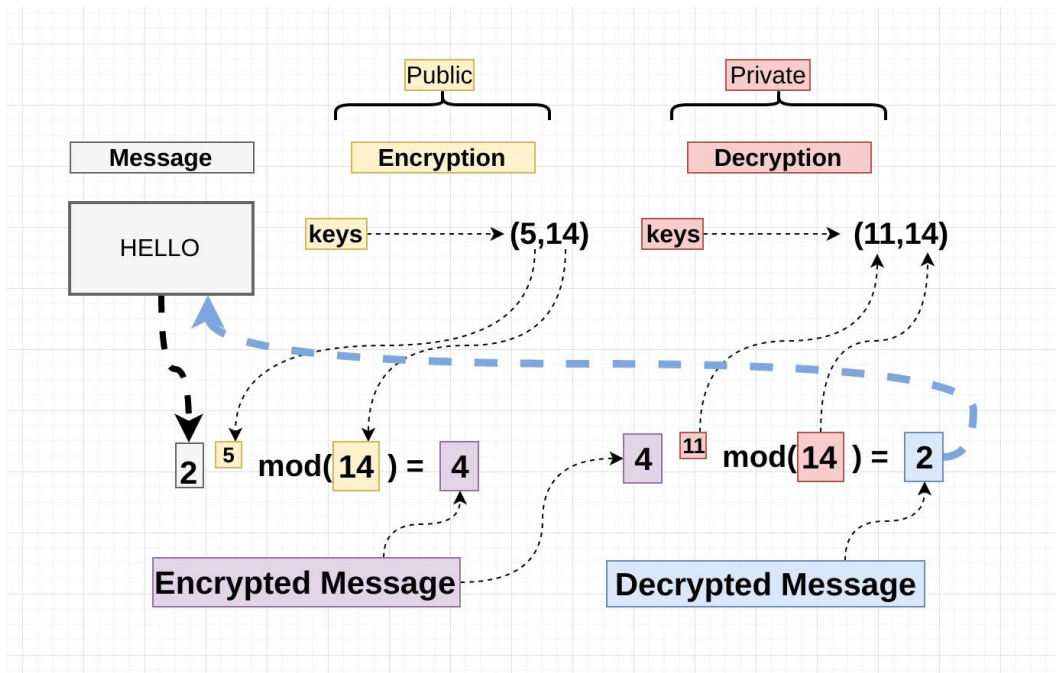


Figure 3.12 RSA working example

RC4 Cipher Details

Key Sizes	1,024 to 4,096 bit typical
Block Size	Depends upon key size
Type	Asymmetric
Rounds	1

3.6 Additional Algorithm Approach

3.6.1 Layered Encryption

The given approach also uses the given three encryption techniques namely AES, DES and RC4. In this given approach a file is selected and then encrypted with one of the following three algorithms DES and then this encrypted message, say m1, is again encrypted using another encryption technique RC4 and this encrypted message, say m2, is further encrypted using AES to get the final encrypted file which is then uploaded to the cloud.

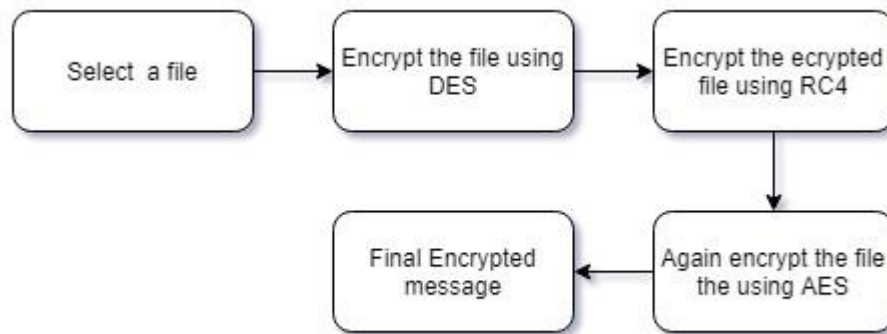


Figure 3.13 Layered Encryption

For the decryption part, the file is downloaded from the cloud and then decrypted firstly using AES and then the decrypted file is again decrypted using RC4 and lastly using DES to get the original message.

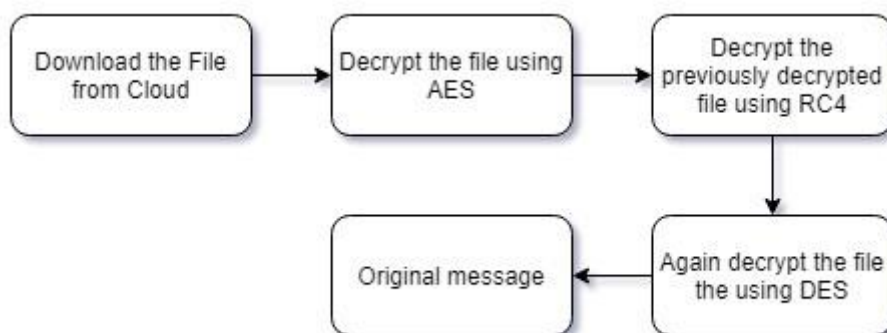


Figure 3.14 Layered Decryption

3.6.2 Hybrid Cryptosystem

Hybrid Cryptosystem combines the convenience of the public-key cryptosystem with the efficiency of a symmetric key cryptosystem. Asymmetric encryption techniques are convenient as they do not require the sender and the receiver to share the secret keys but are generally inefficient in comparison to the symmetric key cryptosystem.

Hybrid cryptosystem uses both symmetric as well as asymmetric encryption techniques. In this approach secret key is encrypted using the asymmetric encryption, RSA, and the file data is encrypted using the symmetric encryption techniques, AES.

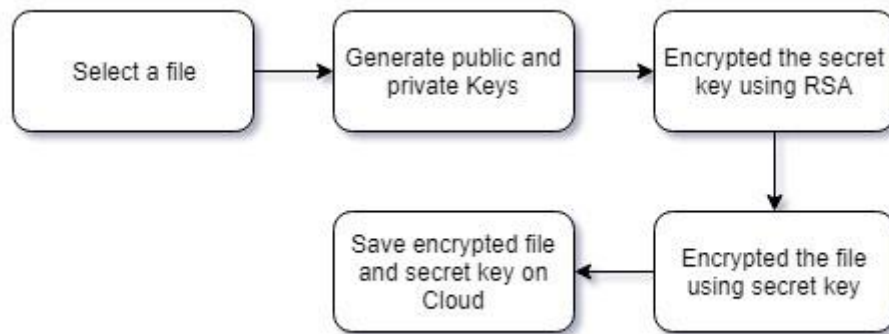


Figure 3.15 Encryption using Hybrid Cryptosystem

Decryption is done by downloading the file from the cloud and then the encrypted key is decrypted using RSA and then this secret key is used to decrypt the encrypted file.

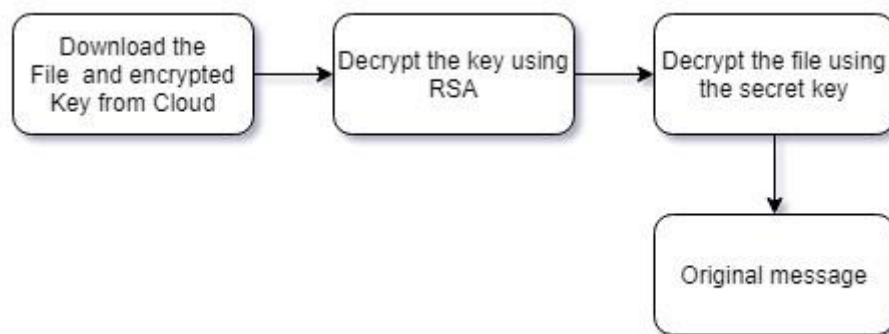
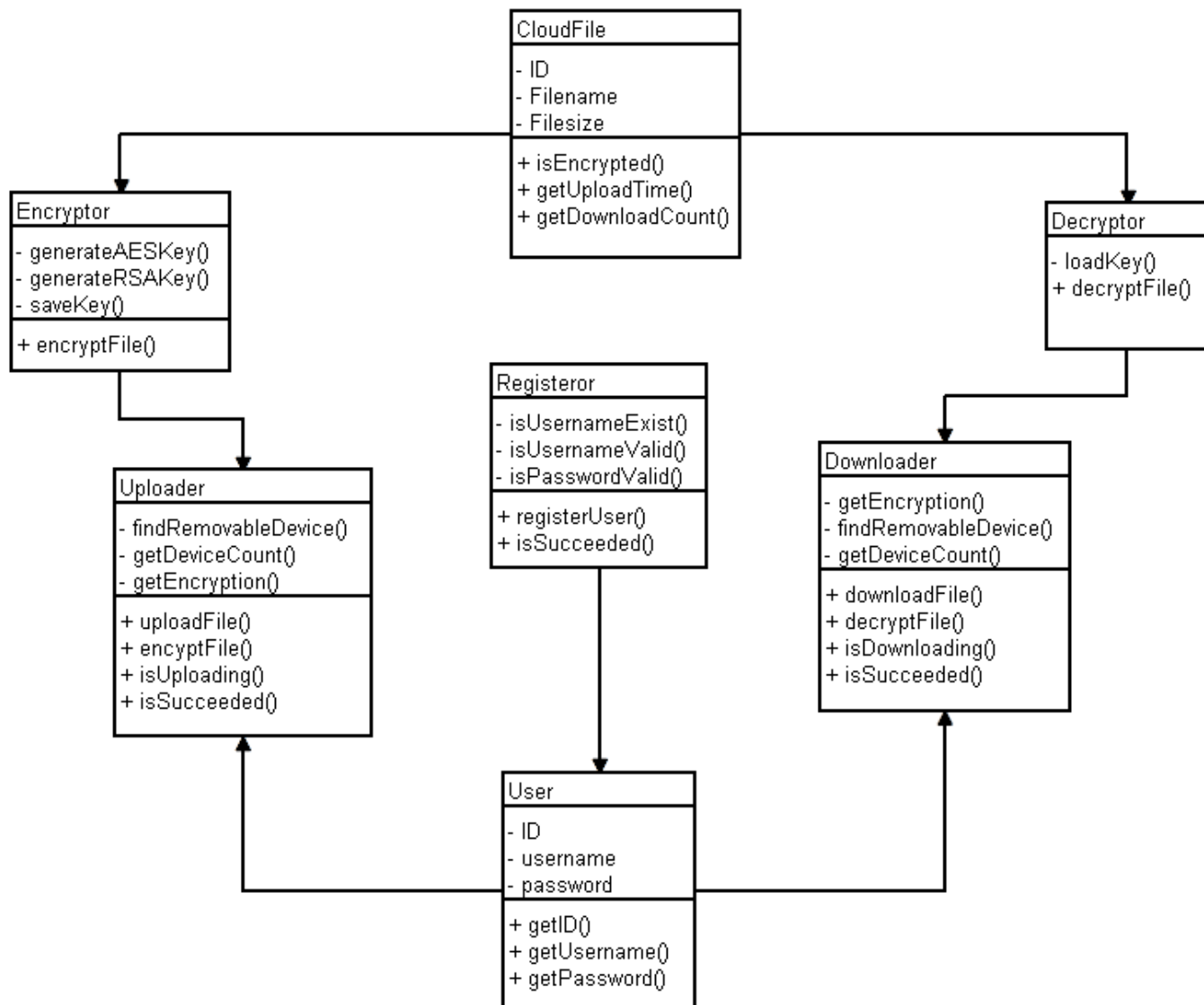


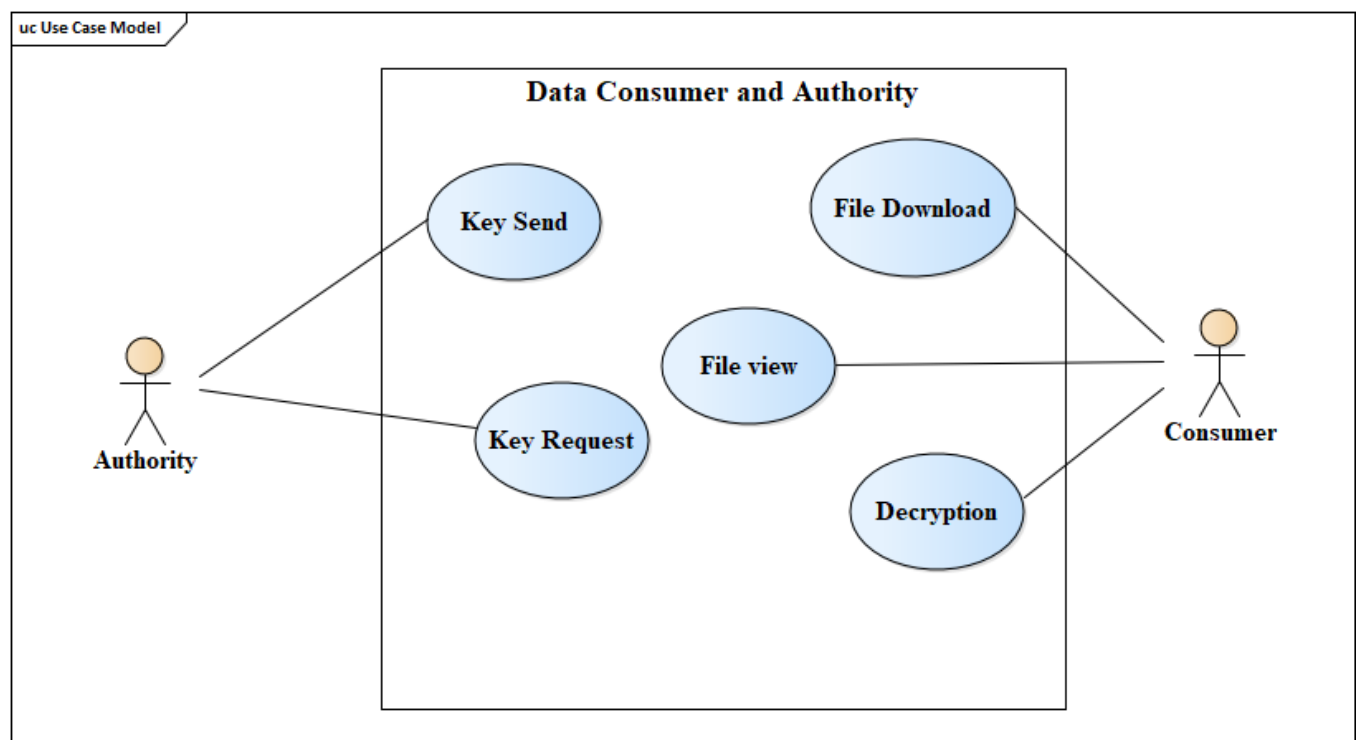
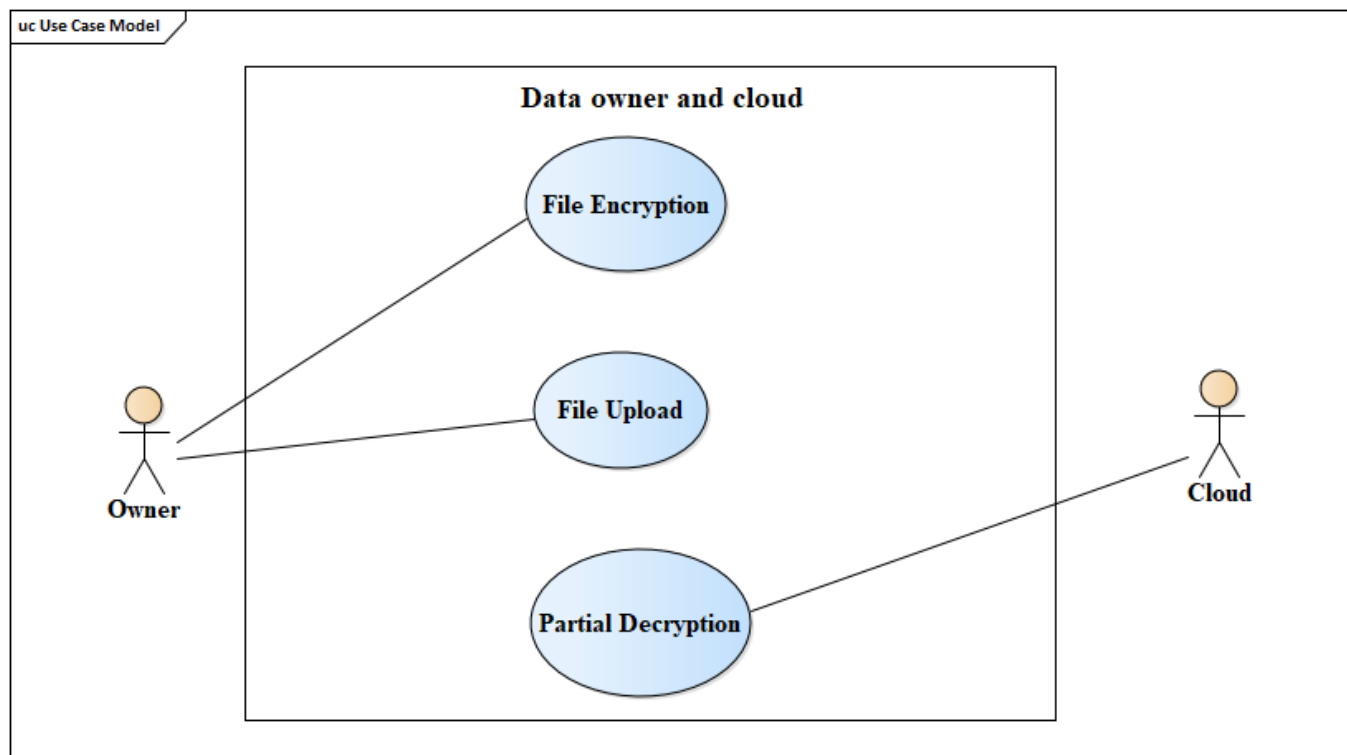
Figure 3.16 Decryption using Hybrid Cryptosystem

4. System Diagrams

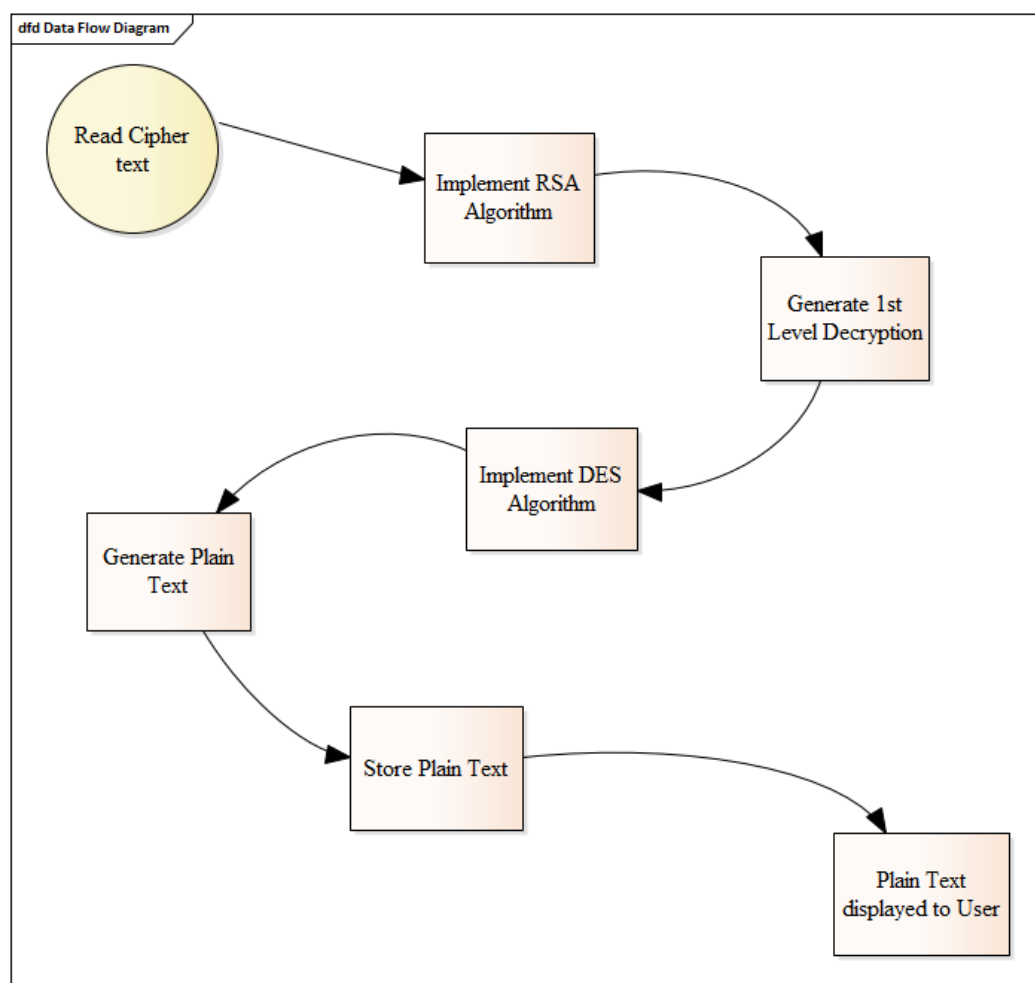
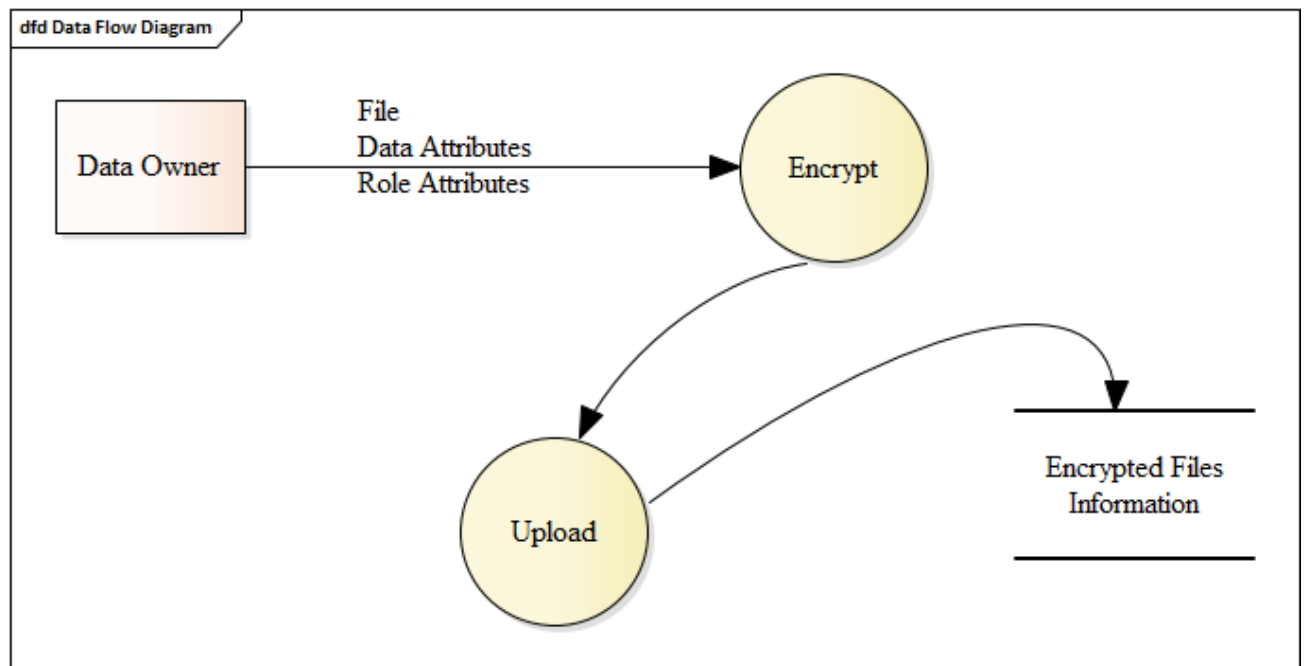
4.1 Class Diagram



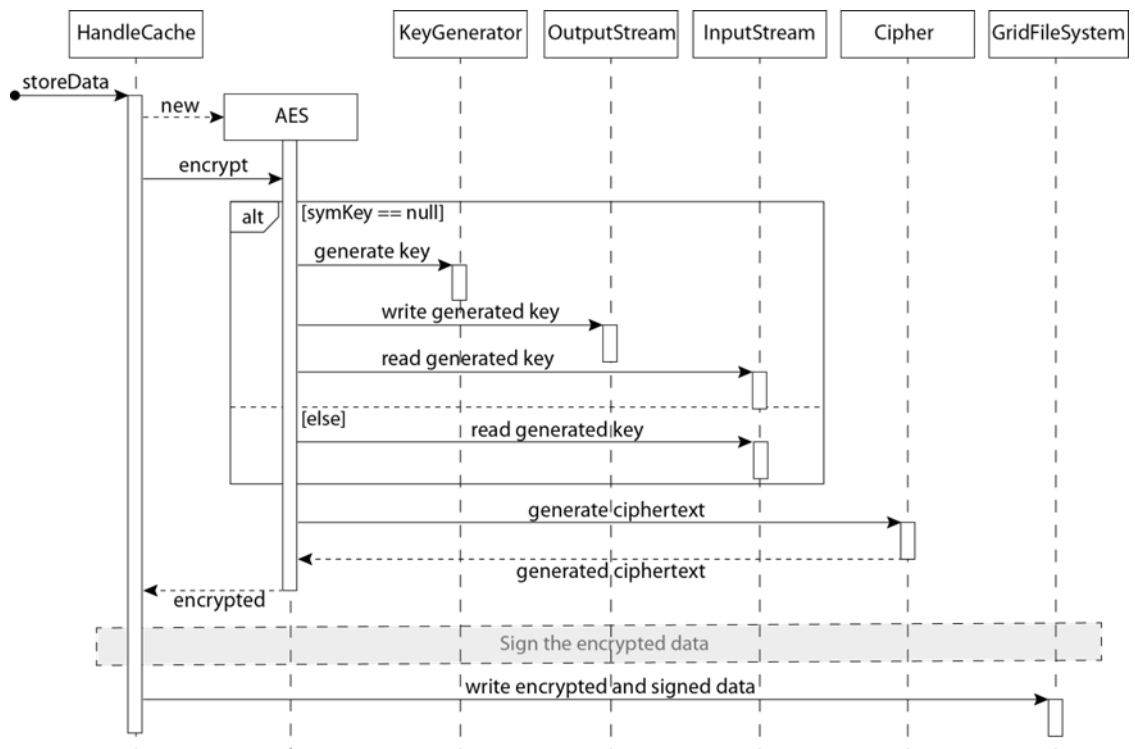
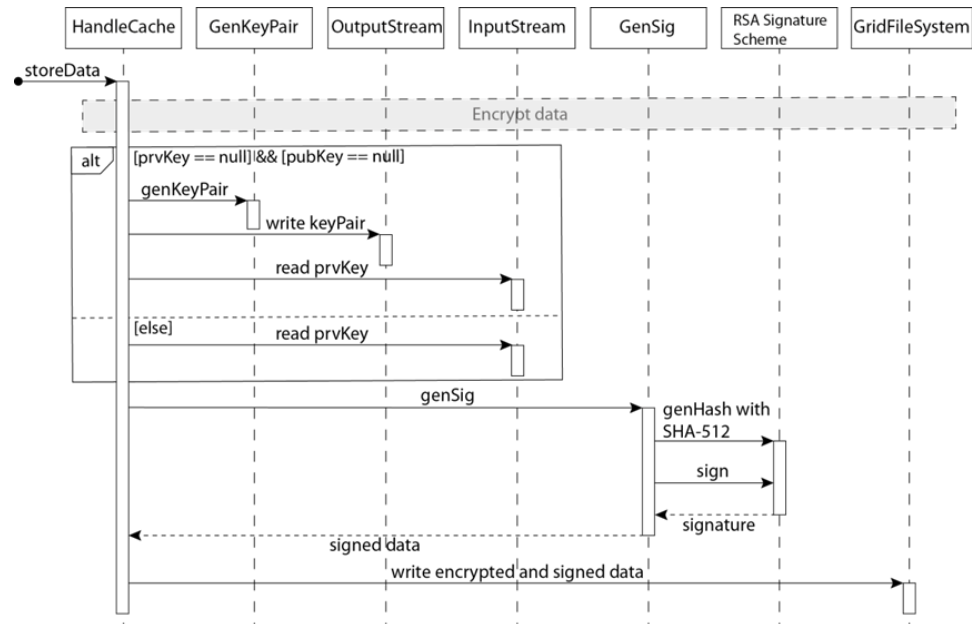
4.2 Use Case Diagrams



4.3 Data Flow Diagrams



4.4 Sequence Diagrams



5. Test Plans

5.1 Test Files

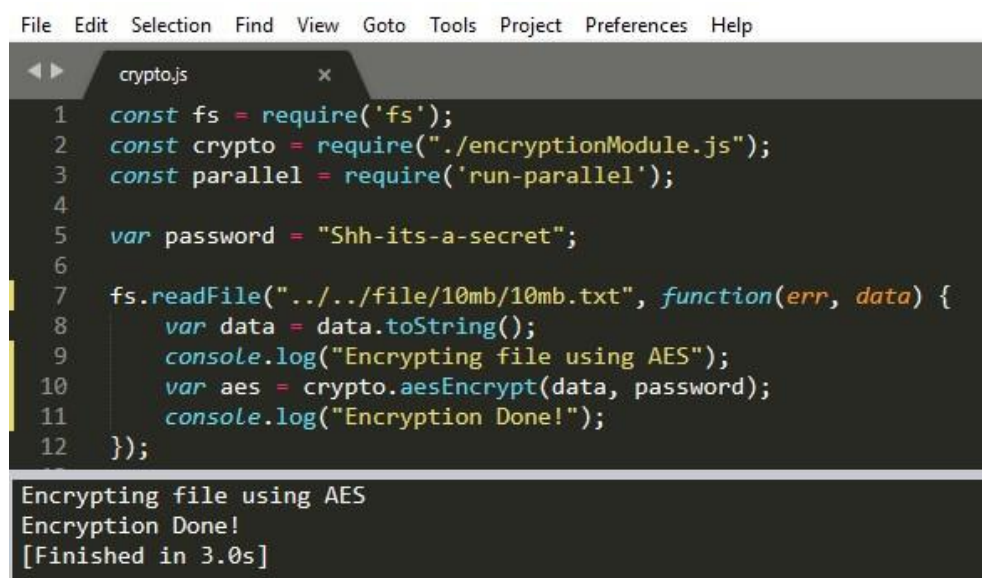
Testing will be done on different files varying from *1mb* to *30mb*. Timing for encryption as well as decryption of such files using hybrid algorithm will be noted and compared with the respective encryption and decryption timings of same files using AES encryption alone. Similarly, the hybrid encryption approach will be compared rest of the two approaches namely layered encryption and hybrid cryptosystem.

5.2. Processing Time

This section provides a rough idea of the comparison between the standard algorithm and the proposed algorithm. A file of size 10mb is taken and is encrypted, firstly, AES and then using the hybrid algorithm, layered algorithm and hybrid cryptosystem. Also, decryption is done of the encrypted file using both, above mentioned, algorithms. The snippet of each algorithm outlines the time required to process the algorithm.

5.2.1 Encryption and Decryption Time for AES

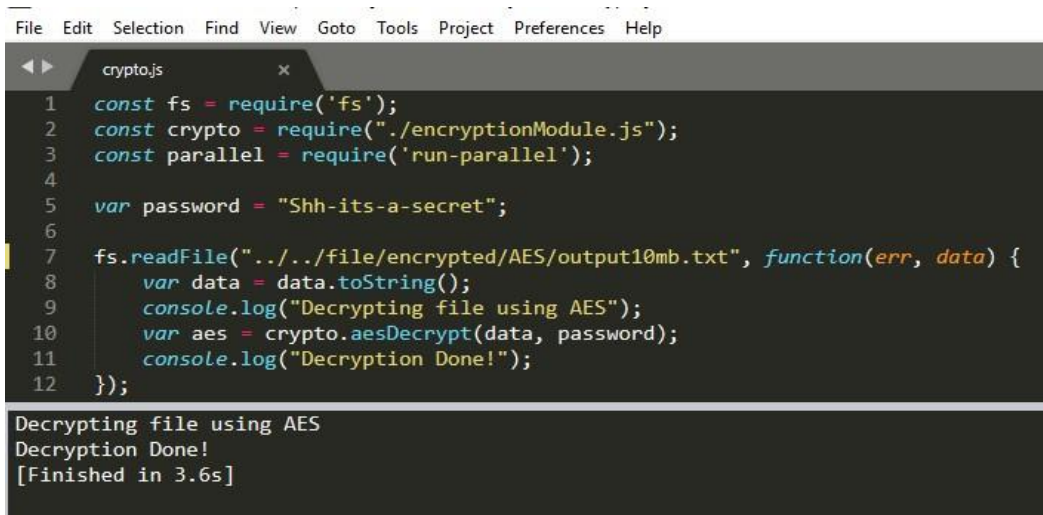
A file named *file.txt* of size 10mb is encrypted using the AES algorithm. Figure 5.1 shows how much time is required to encrypt the file while figure 5.2 depicts the decryption time.



```
File Edit Selection Find View Goto Tools Project Preferences Help
crypto.js
1  const fs = require('fs');
2  const crypto = require('./encryptionModule.js');
3  const parallel = require('run-parallel');
4
5  var password = "Shh-its-a-secret";
6
7  fs.readFile("../..file/10mb/10mb.txt", function(err, data) {
8      var data = data.toString();
9      console.log("Encrypting file using AES");
10     var aes = crypto.aesEncrypt(data, password);
11     console.log("Encryption Done!");
12 });
```

```
Encrypting file using AES
Encryption Done!
[Finished in 3.0s]
```

Figure 5.1 Time to encrypt 10mb file using AES



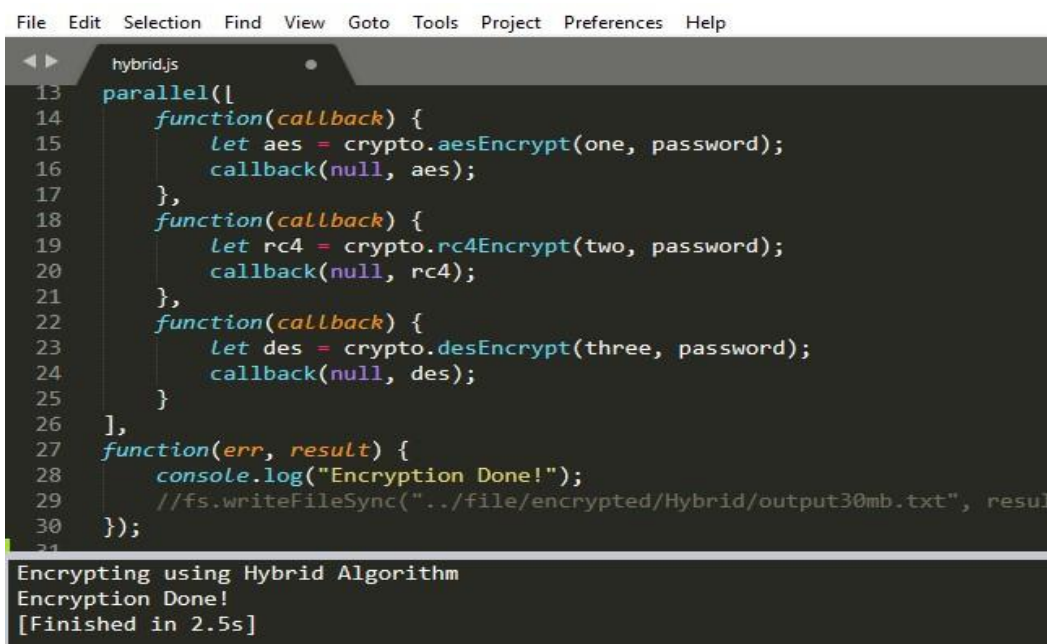
```
File Edit Selection Find View Goto Tools Project Preferences Help
crypto.js
1 const fs = require('fs');
2 const crypto = require("./encryptionModule.js");
3 const parallel = require('run-parallel');
4
5 var password = "Shh-its-a-secret";
6
7 fs.readFile("../file/encrypted/AES/output10mb.txt", function(err, data) {
8   var data = data.toString();
9   console.log("Decrypting file using AES");
10  var aes = crypto.aesDecrypt(data, password);
11  console.log("Decryption Done!");
12 });
```

```
Decrypting file using AES
Decryption Done!
[Finished in 3.6s]
```

Figure 5.2 Time to decrypt 10mb of file using AES

5.2.2 Encryption and Decryption Time for Hybrid Algorithm

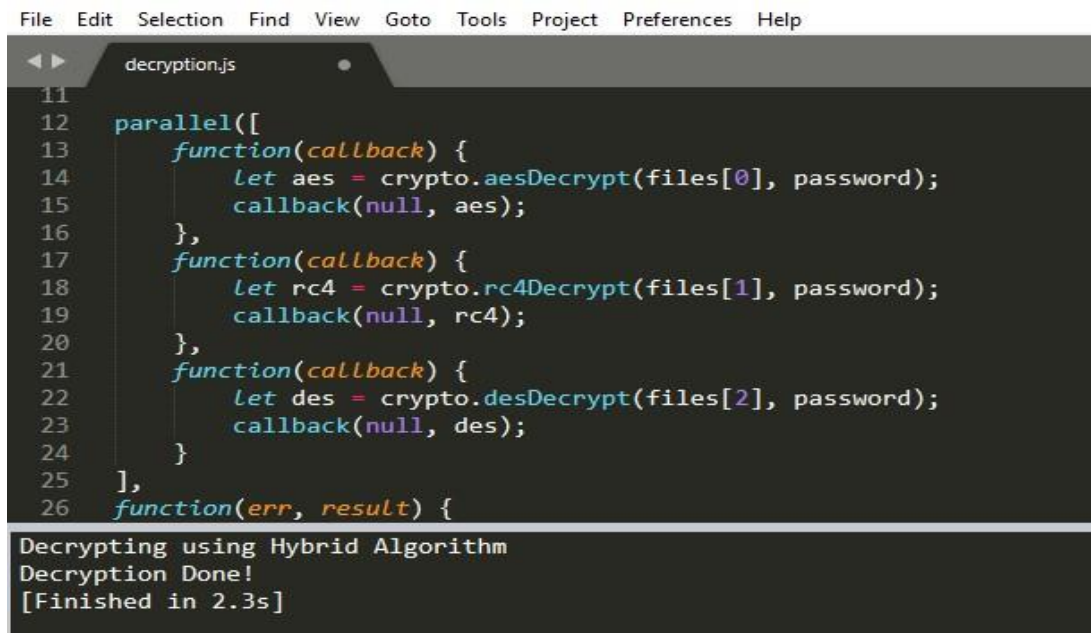
Now the file is encrypted using the hybrid algorithm (AES, DES, RC4). Figure 5.3 shows the time taken to encrypt the file while figure 5.4 shows the decryption time required.



```
File Edit Selection Find View Goto Tools Project Preferences Help
hybrid.js
13 parallel([
14   function(callback) {
15     let aes = crypto.aesEncrypt(one, password);
16     callback(null, aes);
17   },
18   function(callback) {
19     let rc4 = crypto.rc4Encrypt(two, password);
20     callback(null, rc4);
21   },
22   function(callback) {
23     let des = crypto.desEncrypt(three, password);
24     callback(null, des);
25   }
26 ],
27 function(err, result) {
28   console.log("Encryption Done!");
29   //fs.writeFileSync("../file/encrypted/Hybrid/output30mb.txt", result);
30 });
```

```
Encrypting using Hybrid Algorithm
Encryption Done!
[Finished in 2.5s]
```

Figure 5.3 Time to encrypt 10mb file using Hybrid Algorithm



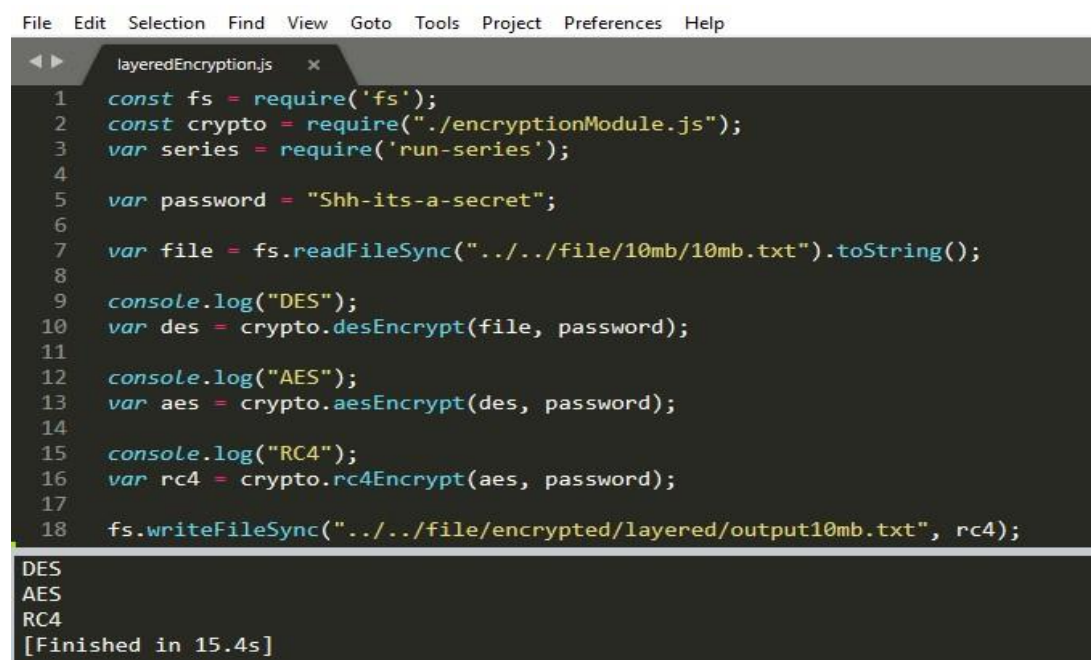
```
File Edit Selection Find View Goto Tools Project Preferences Help
deryption.js
11
12 parallel([
13   function(callback) {
14     let aes = crypto.aesDecrypt(files[0], password);
15     callback(null, aes);
16   },
17   function(callback) {
18     let rc4 = crypto.rc4Decrypt(files[1], password);
19     callback(null, rc4);
20   },
21   function(callback) {
22     let des = crypto.desDecrypt(files[2], password);
23     callback(null, des);
24   }
25 ],
26 function(err, result) {
  Decrypting using Hybrid Algorithm
  Decryption Done!
  [Finished in 2.3s]
```

Figure 5.4 Time to decrypt 10mb file using Hybrid Algorithm

5.2.3 Encryption and Decryption Time for Layered Algorithm

The file is encrypted using the layered algorithm with one over the other (AES, DES, RC4).

Figure 5.5 shows the time taken to encrypt the file while figure 5.6 shows the decryption time required.



```
File Edit Selection Find View Goto Tools Project Preferences Help
layeredEncryption.js x
1  const fs = require('fs');
2  const crypto = require('./encryptionModule.js');
3  var series = require('run-series');
4
5  var password = "Shh-its-a-secret";
6
7  var file = fs.readFileSync(".././file/10mb/10mb.txt").toString();
8
9  console.log("DES");
10 var des = crypto.desEncrypt(file, password);
11
12 console.log("AES");
13 var aes = crypto.aesEncrypt(des, password);
14
15 console.log("RC4");
16 var rc4 = crypto.rc4Encrypt(aes, password);
17
18 fs.writeFileSync(".././file/encrypted/layered/output10mb.txt", rc4);
  DES
  AES
  RC4
  [Finished in 15.4s]
```

Figure 5.5 Time to encrypt 10mb file using Layered Algorithm

```

File Edit Selection Find View Goto Tools Project Preferences Help
layeredDecryption.js x
1 const fs = require('fs');
2 const crypto = require('./encryptionModule.js');
3 var series = require('run-series');
4
5 var password = "Shh-its-a-secret";
6
7 var file = fs.readFileSync("../file/encrypted/layered/output10mb.txt").toString();
8
9 console.log("RC4");
10 var rc4 = crypto.rc4Decrypt(file, password);
11
12 console.log("AES");
13 var aes = crypto.aesDecrypt(rc4, password);
14
15 console.log("DES");
16 var des = crypto.desDecrypt(aes, password);

```

```

RC4
AES
DES
[Finished in 16.9s]

```

Figure 5.6 Time to decrypt 10mb file using Layered Algorithm

5.2.4 Encryption and Decryption Time for Hybrid Cryptosystem

Secret key is encrypted using RSA and the file is encrypted using AES. Figure 5.7 shows the time taken to encrypt the file while figure 5.8 shows the decryption time required.

```

File Edit Selection Find View Goto Tools Project Preferences Help
main.js x
11
12 rsa.encrypt(
13   msg,
14   publicJwk,
15   'SHA-256',
16   ).then( (encrypted) => {
17     // now you get an encrypted message in Uint8Array
18     var obj = {data : encrypted};
19     // console.log(encrypted);
20     fs.writeFileSync('encrypt.json', JSON.stringify(obj));
21   });
22
23 fs.readFile("../file/10mb/10mb.txt", function(err, data) {
24   var data = data.toString();
25   console.log("Encrypting file using AES");
26   var aes = crypto.aesEncrypt(data, password);
27   console.log("Encryption Done!");
28 });

```

```

Encrypting file using AES
Encryption Done!
[Finished in 3.7s]

```

Figure 5.7 Time to encrypt 10mb file using Hybrid Cryptosystem

```
16 ▼ rsa.decrypt(  
17     encrypted,  
18     privateJwk,  
19     'SHA-256',  
20 )  
21 ▼ .then( (decrypted) => {  
22     // now you get the decrypted message  
23     // console.log(new TextDecoder('utf-8').decode(decrypted));  
24     decrypted = new TextDecoder('utf-8').decode(decrypted);  
25 ▼ fs.readFile("../file/encrypted/AES/output10mb.txt", function(err, data) {  
26     var data = data.toString();  
27     console.log("Decrypting file using AES");  
28     var aes = crypto.aesDecrypt(data, password);  
29     console.log("Decryption Done!");  
30 });  
31 });  
32  
222,  
... 156 more items ]  
Decrypting file using AES  
Decryption Done!  
[Finished in 3.8s]
```

Figure 5.8 Time to decrypt 10mb file using Hybrid cryptosystem

5.3 Conclusion

The file of size 10mb was encrypted and decrypted using the standard algorithm as well as the proposed algorithm. The timings from snippets in section 5.2, shows, that the proposed algorithm is both comparatively faster in encryption of the file as well as decryption. The next chapter, Chapter 6, provides a detailed comparison between the standard and proposed algorithm, taking into account files of different sizes, in terms of time required for encryption and decryption.

6. RESULTS AND PERFORMANCE ANALYSIS

6.1 Comparison of Cryptographic Algorithms

In this section, various cryptographic algorithms are compared based on their key lengths, block sizes, number of rounds and other assets.

Table 6.1 Comparison of different Encryption Algorithms

Factors	DES	AES	RC4
Created By	IBM in 1975	Vincent Rijmen, Joan Daemen in 2001	Ron Rivest in 1987
Key Length	56 bits	128, 192 or 256 bits	40 – 2048 bits
Round(s)	16	10,12 or 14	1
Block Size	64 bits	128 bits	2064 bits (1684 effective)
Speed	Slow	Fast	Fast
Security	Not Secure Enough	Excellent Security	Adequate Security

6.2 Algorithms Testing

Testing on various files will be done using AES (Standard Algorithm) and proposed algorithm (Hybrid Algorithm – AES, DES and RC4) so that the performance of the algorithm can be recorded in relation to the increasing size of the file. The files used in the tests vary from 1mb till 30mb. Graphs of both the algorithms along with their tables are shown below.

6.2.1 AES (Advanced Data Standard) Testing

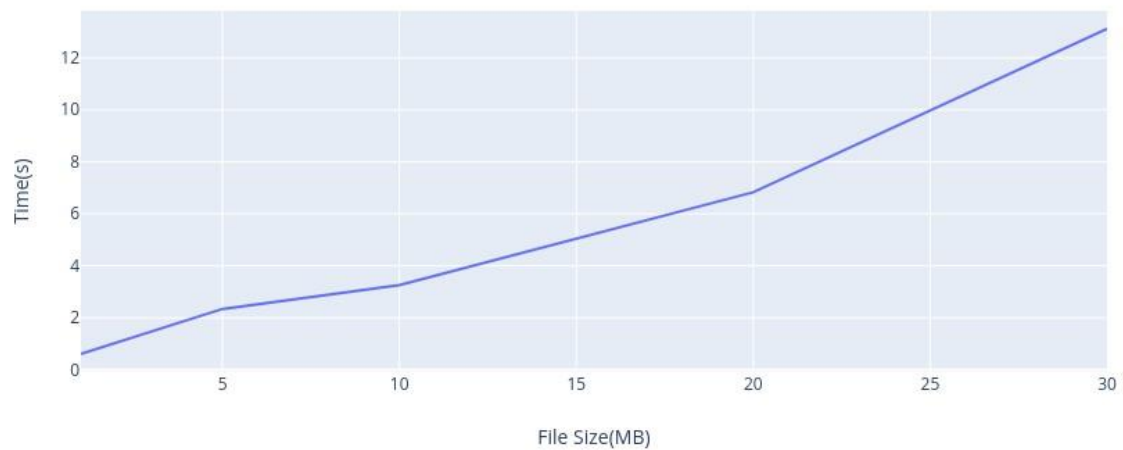
Encryption of file sizes ranging 1mb, 5mb, 10mb, 20mb and 30mb is done using AES Encryption. Encryption time of these files is calculated and plotted on a graph depicted below.

Table 6.2 AES Encryption Time

File Size(MB)	Encryption Time(s)
---------------	--------------------

1 MB	0.617
5 MB	2.344
10 MB	3.257
20 MB	6.829
30MB	13.110

AES Encryption



Graph 6.1 Encryption using AES

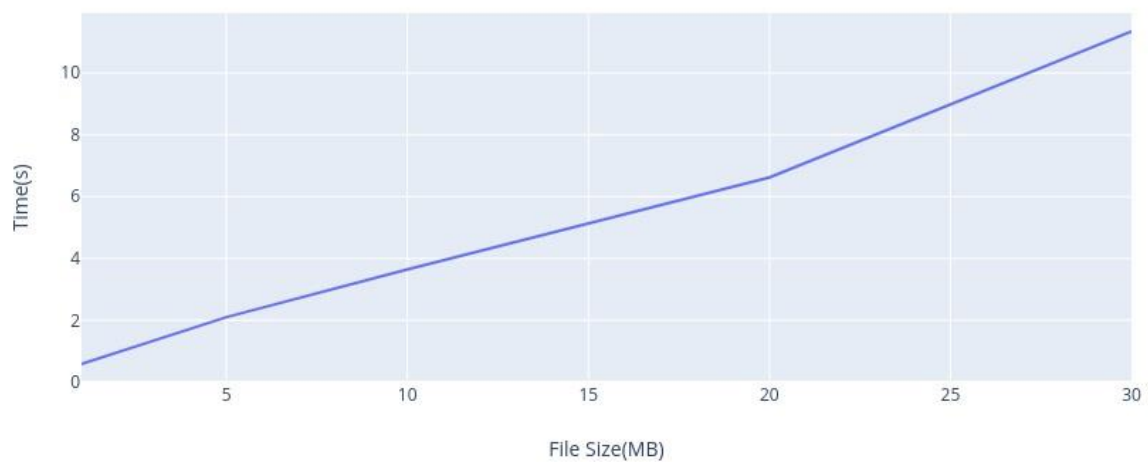
Similarly, decryption of file sizes ranging 1mb, 5mb, 10mb, 20mb and 30mb is done using AES decryption algorithm. Decryption time of these files is calculated and plotted on a graph depicted below.

Table 6.3 AES Decryption Time

File Size(MB)	Decryption Time(s)
1 MB	0.589

5 MB	2.099
10 MB	3.643
20 MB	6.625
30MB	11.345

AES Decryption



Graph 6.2 Decryption using AES

6.2.2 Hybrid Encryption (AES, DES, RC4) Testing

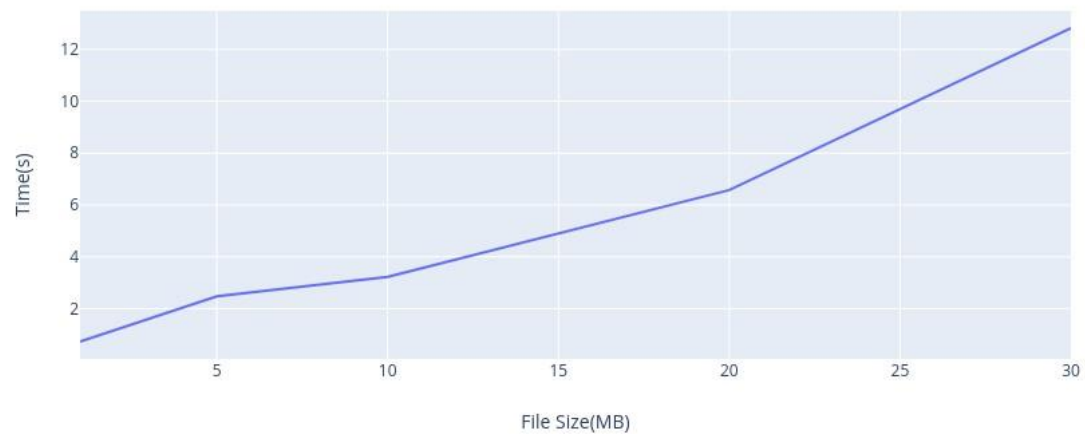
Again, encryption of file sizes ranging 1mb, 5mb, 10mb, 20mb and 30mb is done using Hybrid Encryption. As already done on the above process, encryption time of these files is calculated and plotted on a graph depicted below.

Table 6.4 Hybrid Encryption Time

File Size(MB)	Encryption Time(s)
1 MB	0.732

5 MB	2.476
10 MB	3.221
20 MB	6.575
30MB	12.818

Hybrid Encryption



Graph 6.3 Encryption using Hybrid Algorithm

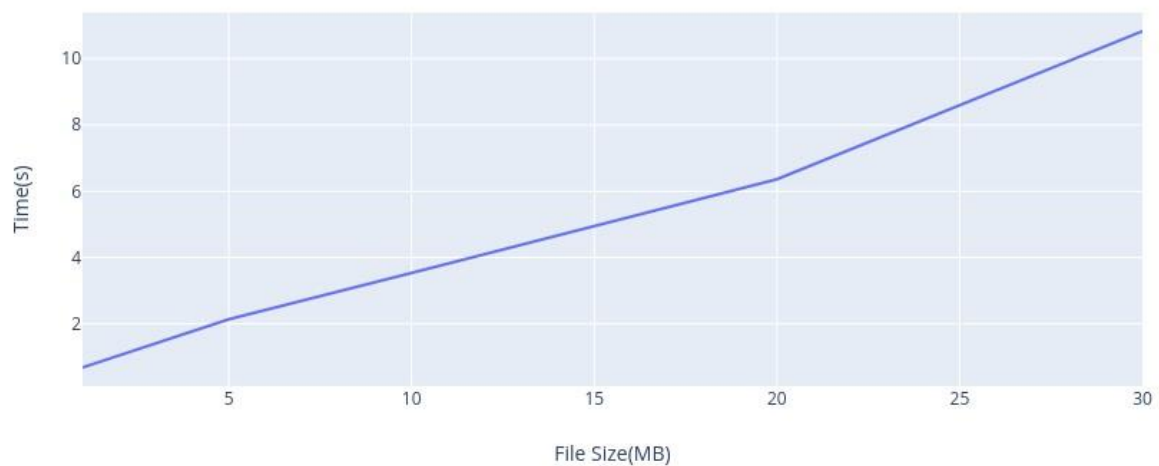
Similarly, decryption of file sizes ranging 1mb, 5mb, 10mb, 20mb and 30mb is done using hybrid decryption algorithm. Decryption time of these files is calculated and plotted on a graph depicted below.

Table 6.5 Hybrid Decryption Time

File Size(MB)	Decryption Time(s)
1 MB	0.685
5 MB	2.134

10 MB	3.532
20 MB	6.363
30MB	10.828

Hybrid Decryption



Graph 6.4 Encryption using Hybrid Algorithm

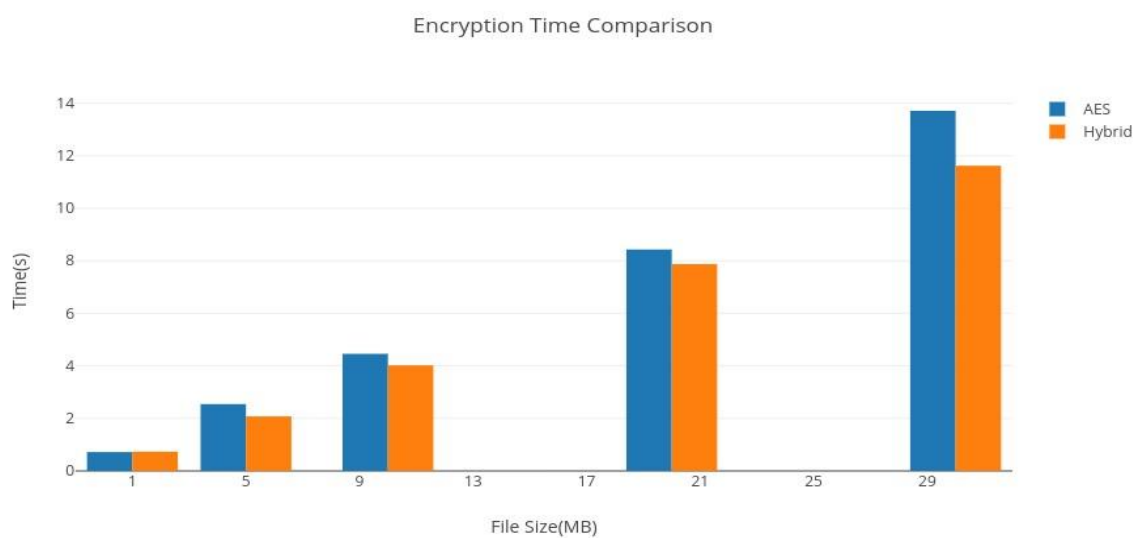
6.3 Comparison of Encryption Algorithms Testing Data

Comparisons of the encryption time as well as decryption time of the files ranging 1mb to 30mb is shown in the table given below. Hybrid algorithm needs 10 to 15 percent less time for file to be encrypted in comparison to other encryption techniques. With single encryption algorithm such kind of data security cannot be provided. Table 6.6 Comparison of Different Algorithms

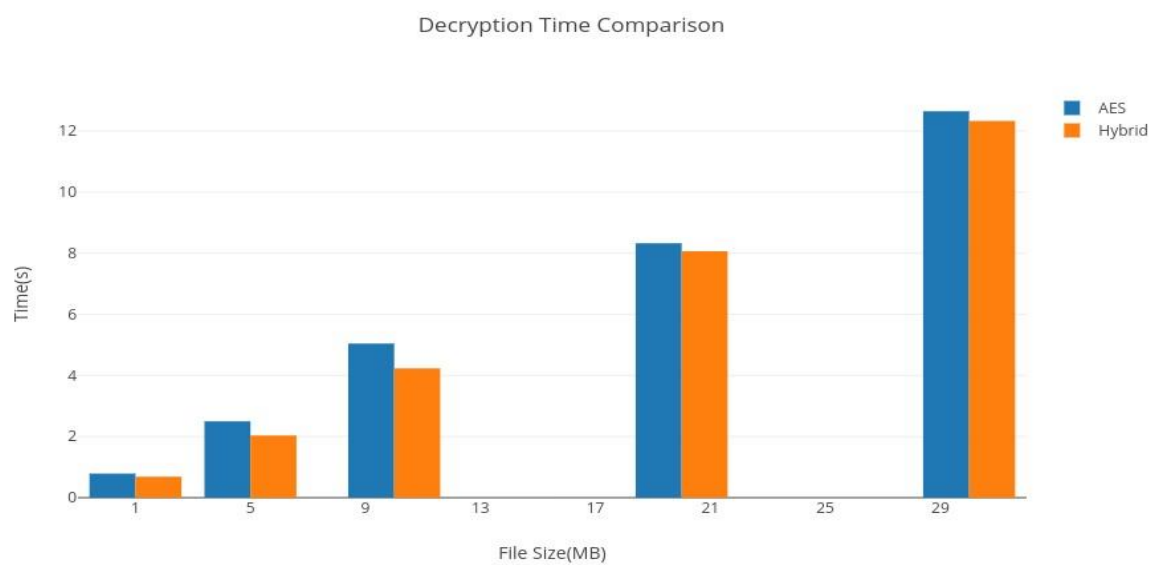
File Size	AES Encryption	Hybrid Encryption	AES Decryption	Hybrid Decryption
1 MB	0.617	0.732	0.589	0.685
5 MB	2.344	2.476	2.099	2.134

10MB	3.257	3.221	3.643	3.532
20MB	6.829	6.575	6.625	6.363
30MB	13.110	12.818	11.345	10.828

Graph below depicts the encryption time of various files.



Graph 6.5 Time Comparison between AES and Hybrid Encryption Algorithm



Graph 6.6 Time Comparison between AES and Hybrid Decryption Algorithm

6.4 Comparison of Hybrid Encryption with Layered Encryption and Hybrid Cryptosystem

Comparisons of the encryption time as well as decryption time of the files ranging 1mb to 30mb of different algorithm approaches is show below. Table 6.7 shows the encryption time whereas table 6.8 shows decryption time.

Table 6.7 Comparison of Different Encryption Approaches

File Size	Hybrid Encryption	Hybrid Cryptosystem	Layered Encryption
1 MB	0.732	0.889	2.032
5 MB	2.476	2.499	7.676
10MB	3.221	3.543	15.121
20MB	6.575	7.125	40.575
30MB	12.818	14.245	57.118

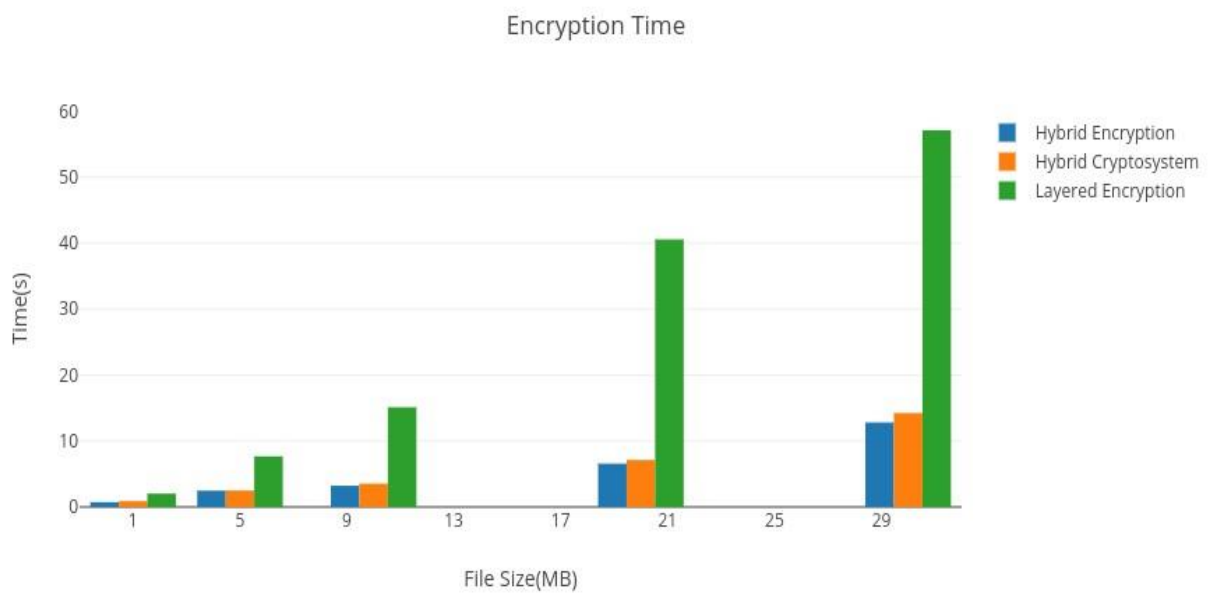
Decryption time for files ranging from 1mb to 30mb using Hybrid algorithm, Layered algorithm and Hybrid Cryptosystem is shown below with hybrid encryption approach clearly being more efficient than the other two approaches.

Table 6.8 Comparison of Different Decryption Approaches

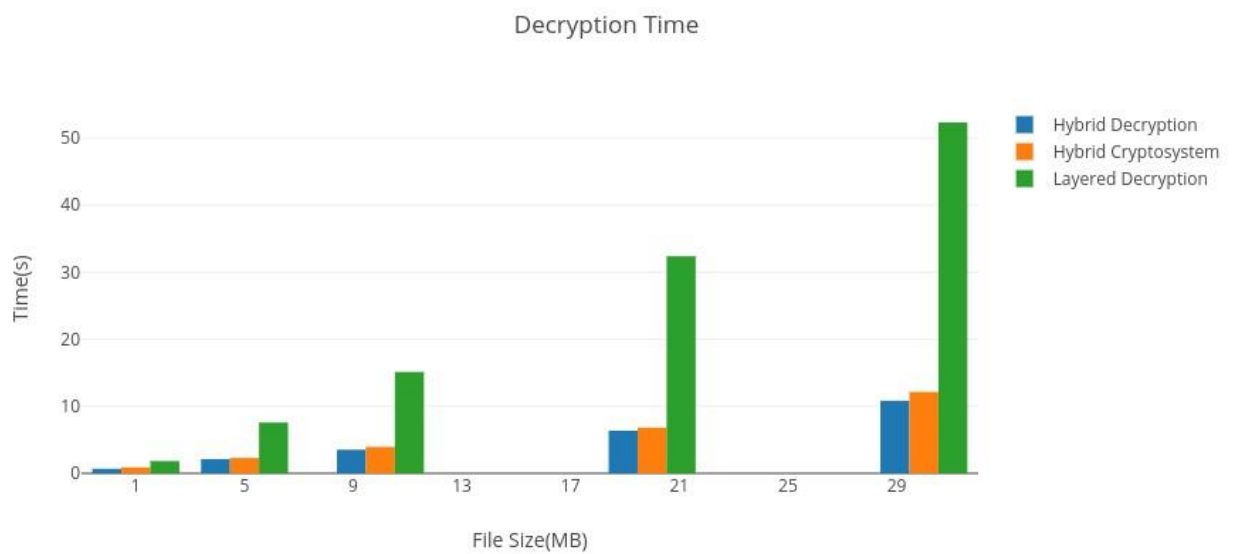
File Size	Hybrid Decryption	Hybrid Cryptosystem	Layered Decryption
1 MB	0.685	0.889	1.832

5 MB	2.134	2.299	7.576
10MB	3.532	3.943	15.121
20MB	6.363	6.825	32.375
30MB	10.828	12.145	52.318

Graph below depicts the encryption time of various files.



Graph 6.7 Comparison between Hybrid, Hybrid Cryptosystem and Layered Encryption



Graph 6.8 Comparison between Hybrid, Hybrid Cryptosystem and Layered Decryption

6.5 Application Screenshots

6.5.1 File Encryption

A user can upload files and store them in an encrypted format.

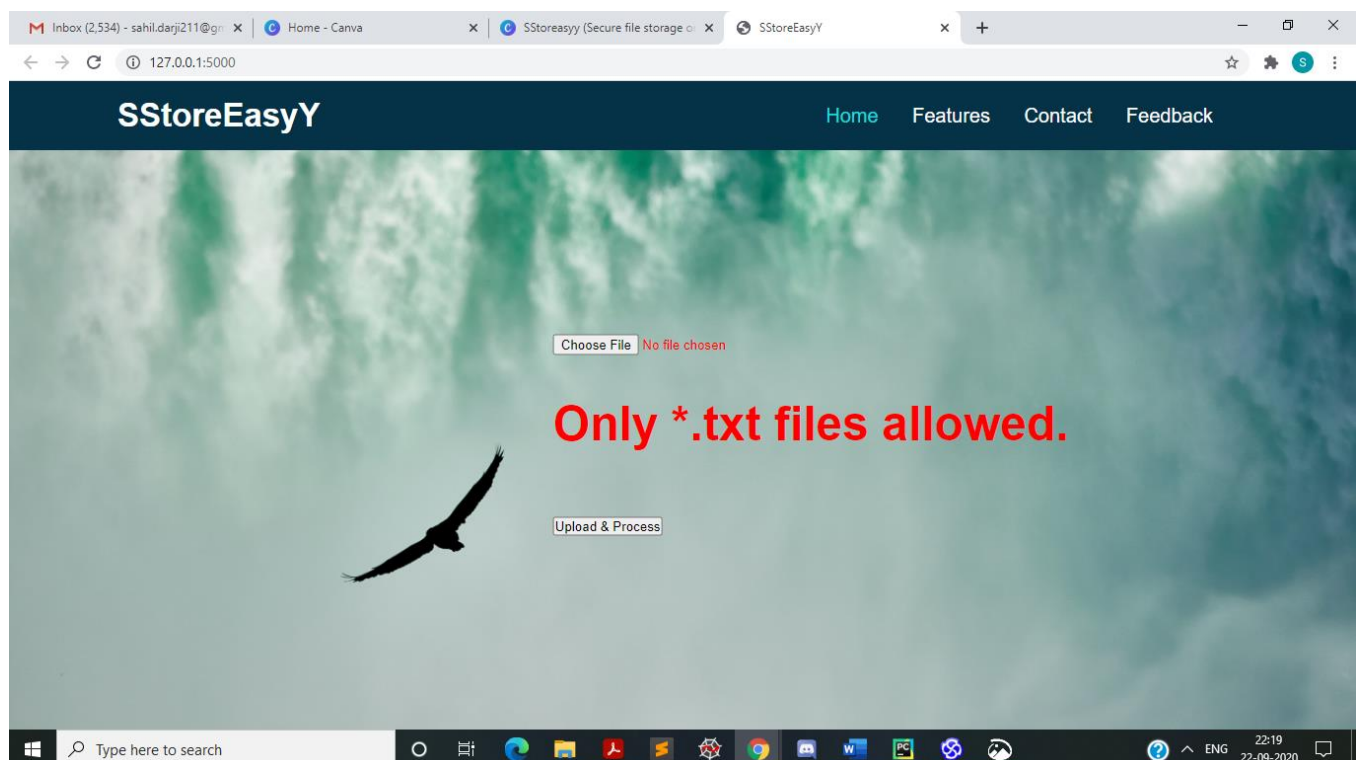


Figure 6.1 File Encryption

6.5.2 Download Key

User can download key for the file which is encrypted and can use it for future.

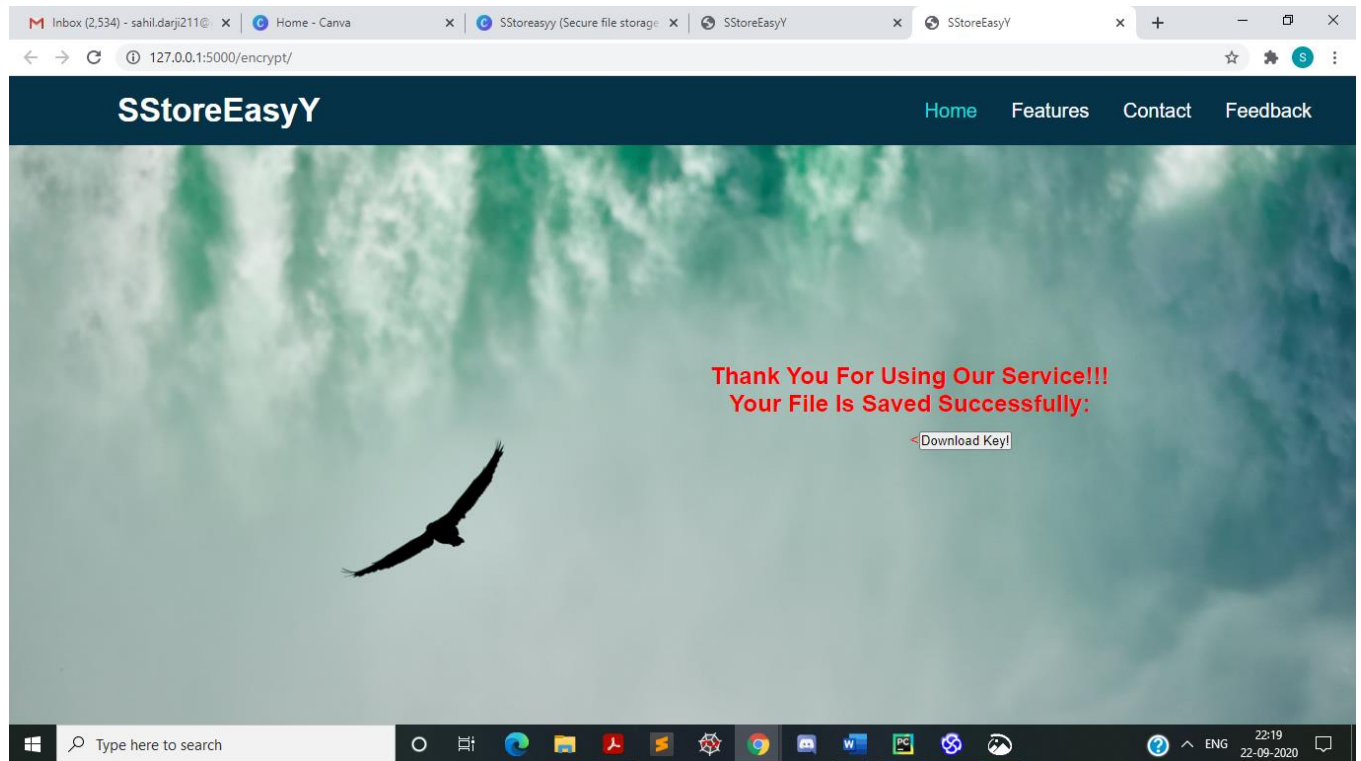


Figure 6.2 Registration Form

6.5.3 Encryption Techniques

The user can select from one of the three encryption algorithm as shown in fig 6.3. Hybrid Algorithm uses three encryption techniques AES, DES and RC4 by splitting a file and using one of the mentioned encryption algorithms. Layered encryption also uses these three algorithms but one over the other. Hybrid Cryptosystem uses asymmetric approach to encrypt the secret key which is used to encrypt the data using symmetric approach.

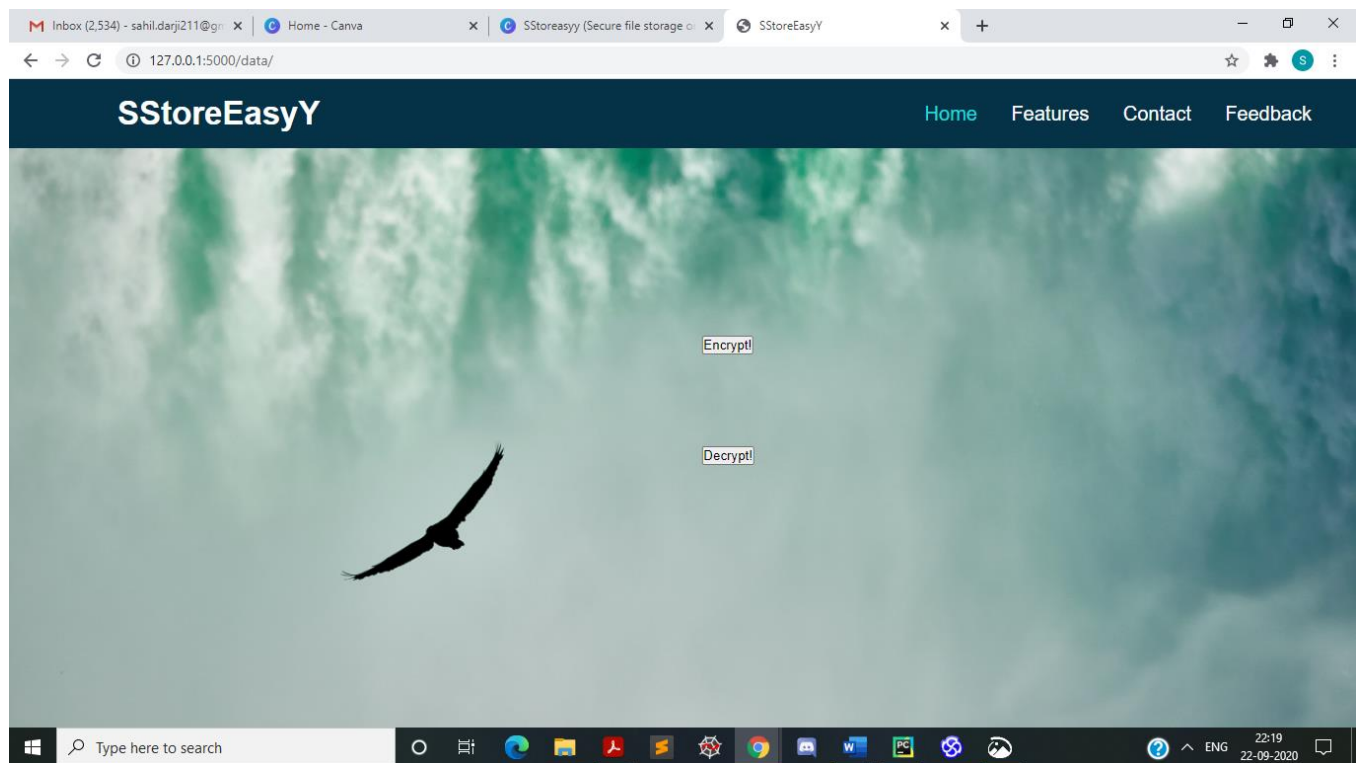


Figure 6.3 Encryption Types

6.5.4 Download File

User can download the encrypted file and get the file in original format.

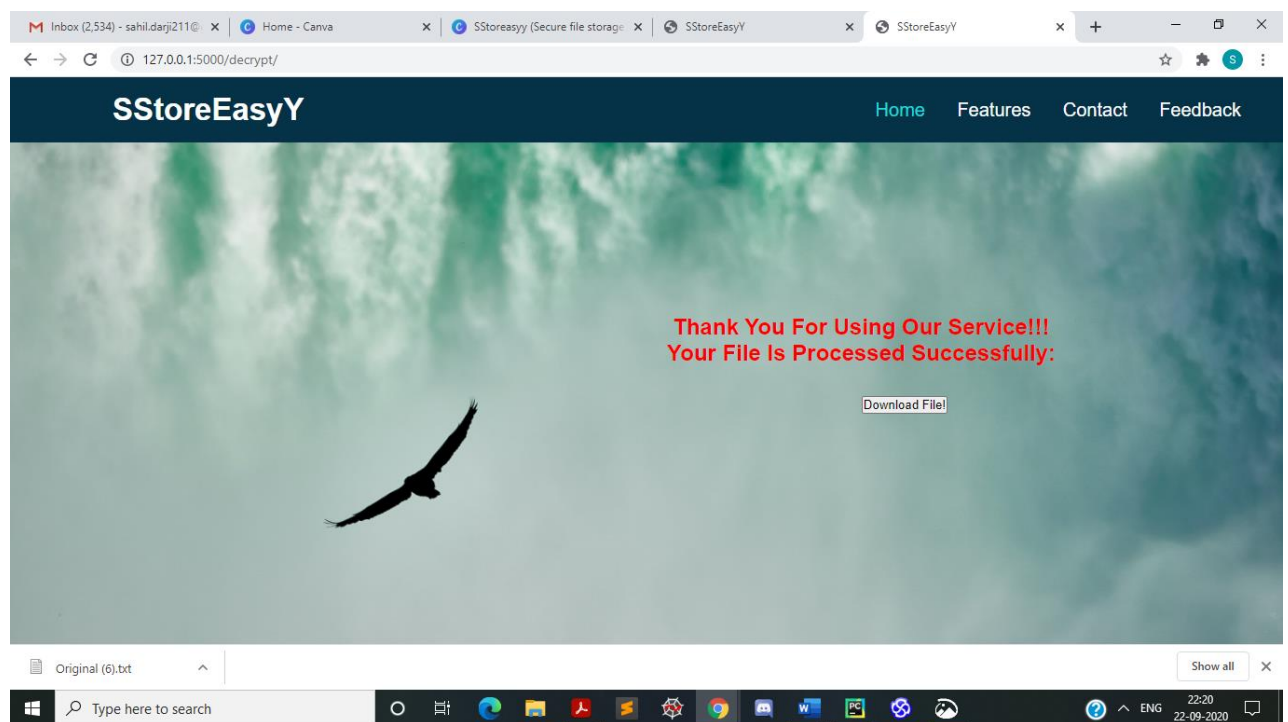


Figure 6.4 Hybrid Encryption

7. CONCLUSIONS

7.1 Conclusions

By the investigation of the accompanying outcomes, we could reason that the hybrid algorithm of AES, DES and RC4 gave us better execution time in contrast with that of AES algorithm alone and also compared to the layered approach and hybrid cryptosystem. We saw that for moderately little document sizes AES individually produced better throughput in bytes per millisecond when contrasted with that of hybrid algorithm including AES, DES and RC4, however, as the measure of record augmented, proposed algorithm indicated better outcomes. Mulling over, the substantial measure of information that business applications will in general store on the cloud, record sizes can fluctuate to extensive numbers, thus the utilization of hybrid algorithm calculation is proposed to actualize staggered security on cloud information storage.

7.2 Applications

We regularly utilize snapchat, Instagram, google drive, our primary concern is, on the off chance that we utilize any informal communication program or information stockpiling we are nearly utilizing distributed computing. We likewise utilize cloud administrations, for example, email offloading that enables a ton in lessening the organizations to cost of advancement and support notwithstanding the enormous advantages of the distributed computing the protection of the information is the greatest worry of the organizations.

In an exploration venture, we are driving our point to give cloud information security and security assurance. Even though distributed computing has numerous points of interest there are yet numerous issues that should be comprehended and in our examination venture, we have made a suite of calculations for versatile control and calculation of encoded information in the cloud. This will go to help the cloud suppliers to control and deal with the physical framework and ensure the information on the cloud stays secure. There can be information vulnerabilities that is susceptible to attackers looking to exploit and assault the information to gain full control over the information and steal it, but the proposed algorithm will help retain the data and provide a complete security over the client's data.

7.3 Future Scope

Based on the examination of execution of single encryption calculations and multiple encryption calculations, we could conclude that hybrid encryption involving multiple encryption algorithms (AES, DES and RC4) provided much better results than single algorithm (AES) implemented alone. In the future, we plan to use certain other encryption algorithms such as Blowfish and other public key encryption techniques like RSA to be implemented in the project. On cloud, we intend to analyse the hybrid algorithm on different execution in real time and against different cryptanalytic assaults with the end goal to check the vigour and unwavering quality of the proposed framework.

This undertaking at that point plans to give better and upgraded ongoing security arrangements with the end goal to give more proficient and improved client involvement with the administrations used on cloud so the issues of information security, defencelessness and nondisavowal can be fathomed.

REFERENCES

- [1] M. Ali *et al.*, “SeDaSC: Secure Data Sharing in Clouds,” in *IEEE Systems Journal*, vol. 11, no. 2, pp. 395-404, June 2017.
- [2] Z. Xiao *et al.*, “Security and Privacy in Cloud Computing,” in *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 843-859, Second Quarter 2013.
- [3] Q. Zhang *et al.*, “Cloud computing: state-of-the-art and research challenges,” *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7-8, 2010.
- [4] M. Armbrust *et al.*, “A view of cloud computing,” *Communications of the ACM*, vol. 53, no. 4, p. 50, Jan. 2010.
- [5] M. Batra *et al.*, “Secure file storage in cloud computing using hybrid encryption algorithm”, *International Journal of Computer Engineering and Applications*, vol. 19, no. 6, June 18. Utilization”, *International Journal of Computer Trends and Technology*, Aug. 2011.

- https://en.wikipedia.org/wiki/Cloud_computing
- https://en.wikipedia.org/wiki/Hybrid_cryptosystem
- <https://ieeexplore.ieee.org/document/8211728>
- <http://www.ijarcs.info/index.php/Ijarcs/article/view/5916>
- <https://www.engpaper.com/cse/secure-file-storage-on-cloud-using-cryptography.html>
- <https://www.ijert.org/review-of-secure-file-storage-on-cloud-using-hybrid-cryptography>