

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 INTRODUCTION TO CLOUD COMPUTING**

Cloud computing has become the current generation buzz in Information Technology industry. Cloud computing has become an alternative to the traditional client-server model based industry. It enables the users to move their data and application software to the network which is different from traditional solutions [1]. The cloud computing is often said as the most promising technology as it has the capacity to change how the entire IT enterprise field works. Cloud computing is nothing but delivering computing as a service based on the demand of the user or client on pay per use model, here the term computing refers to application, software and platform. Cloud computing become famous when amazon lend their server space for the outside world at a moderate cost. This makes way for plenty of company that could not able to afford separate hardware and software for their business to make use of cloud computing. Cloud computing cuts the cost of the company by a greater number as the company needs to pay based on the resource they use. Most of the company started providing storage as a service as the storage business becomes a real big deal to earn profit. Both the cloud service provider and the cloud user gain a lot from this process. From the introduction of the term “cloud” in IT industry, it dominates the trends in IT industry. It is listed as one of the promising technology called as SMAC by IBM. Cloud computing changes the way the computing takes places in this century. Cloud computing is in the middle of the technology era with its features making changes in every industry it reaches. Some consider that cloud computing is marketing hype that puts new face on old technologies such as utility computing, virtualization, or grid computing. With its advantage, cloud computing dominates the IT industry.

#### **1.1.1 NEED FOR CLOUD COMPUTING**

Cloud computing greatly needed in IT industry in order to simplify the work and reduce the cost on storage and other application. It greatly reduces the storage cost, if a startup company needs space for storage they have to buy new expensive server in order to store their data safely, by using cloud computing these startup company can buy storage space from the cloud service

provider based on their demand or usage and they can pay only for what they use. Similarly cloud computing provides a lot of choice to the user on what kind of software they wish to use, the user can check and use the software they wished and pay for it only. It has become a comfort to use cloud computing for the variety of service they provide.

### 1.1.2 ARCHITECTURE OF CLOUD COMPUTING

Cloud computing consists of loosely coupled components that are interconnected with each other for the process of communication between the components. These components combined to work together as cloud computing [2].

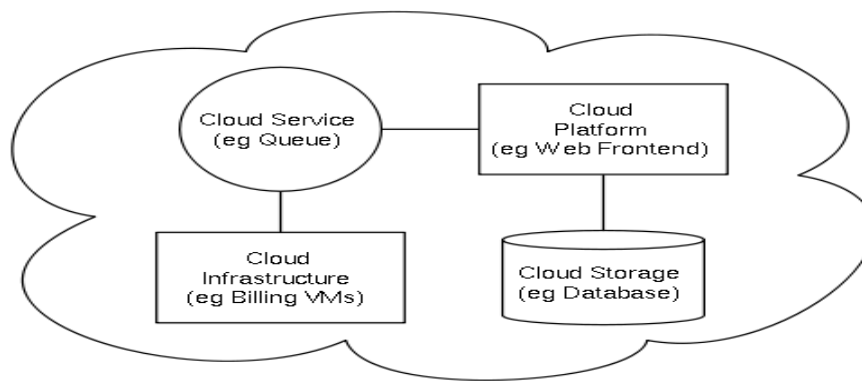


Fig 1.1.Cloud computing architecture [3]

### 1.1.3 WORKING OF CLOUD COMPUTING

Cloud computing are proprietary technology whereas grid computing are based on open source technologies. So only the cloud service provider knows exactly how data, requests, security arrangements are managed. To understand the working of cloud computing, cloud computing has multilayer called front end layer and back end layer. The software that makes interaction with the user is the frontend and the backend consists of hardware and software architecture that delivers the data to the frontend.

The three main cloud service delivery models are

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

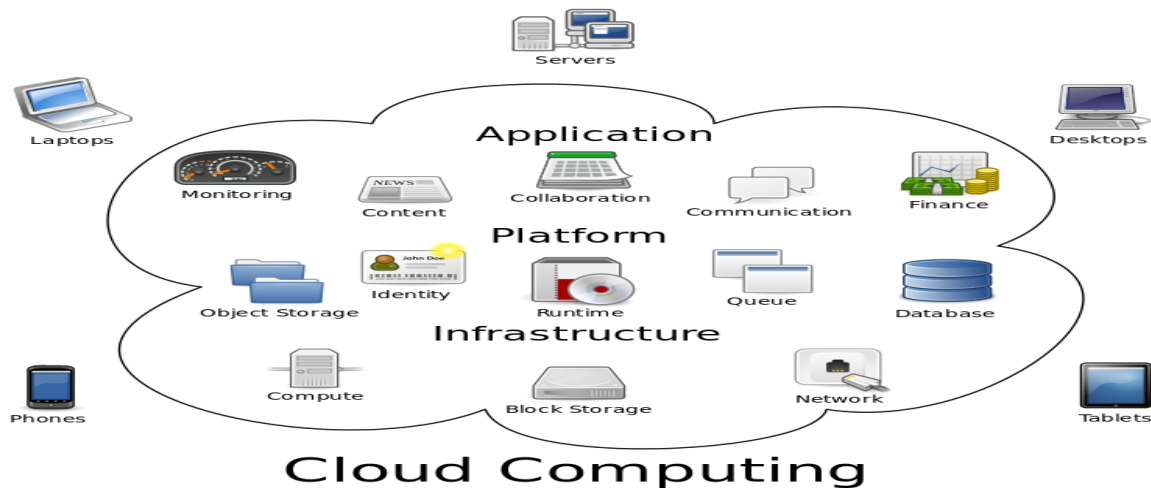


Fig 1.2 cloud computing deployment model [3]

### 1.1.4 APPLICATION OF CLOUD COMPUTING

Cloud computing offers various application which provides the user with lot of choice to make with the software and to reduce the cost. Some of the advantages are listed below,

*File storage:* Cloud computing offers the possibility of storing, accessing and retrieving the data from any of the web enabled interface. It is very simple to use and it provides high speed, availability, scalability for the user. In this scenario, organizations are only paying for the amount of storage they are actually consuming, and without the worries of the daily maintenance of the storage infrastructure.

*Test and Development:* The best scenario for the use of a cloud is a test and development environment. This enables securing a budget, setting up environment through physical assets, significant manpower and time. Then comes the installation and configuration of the platform.

All this can often extend the time it takes for a project to be completed. With cloud computing, there are now readily available environments tailored for needs at fingertips. This often combines, but is not limited to, automated provisioning of physical and virtualized resources [4].

*Big Data analytics:* One of the aspects offered by cloud computing is the ability to tap into vast quantities of both structured and unstructured data to benefit of extracting business value. Retailers and suppliers are now extracting information derived from consumers' buying patterns to target their advertising and marketing campaigns to a particular segment of the population. Social networking platforms are now providing the basis for analytics on behavioral patterns that organizations are using to derive meaningful information. This techniques is provided by the cloud computing.

### **1.1.5 ADVANTAGES OF CLOUD COMPUTING**

The main advantages of using cloud computing are,

*Reliable:* Cloud gives a highly reliable environment where replacement can be done faster. They are having up time of 24\*7 which is impossible in the traditional way.

*Inexpensive:* Cloud is not expensive as there are many tenants or client in the same cloud. So it is very less expense rather than maintaining the own servers.

*Scalable:* If the client wants to expand the business, they can easily improve their business using the cloud methodology.

*Flexible:* Client can easily switch between the servers and they can expand their plan and package when needed based on their needs.

### **1.1.6 ISSUES IN CLOUD COMPUTING**

There are several issues that are associated with cloud computing, some of the issues are  
*Security:* The vendor for Cloud must make sure that the customer does not face any problem such as loss of data or data theft. There is also a possibility where a malicious user can penetrate the cloud by impersonating a legitimate user, there by infecting the entire cloud thus affecting

many customers who are sharing the infected cloud. There are several issues within security in cloud computing. The recent happening in the IT world includes lot of attacks on the data that are stored in the cloud that had directly affected security of the cloud. Cloud computing though provides greater benefits to the user, it gets affected when comes to the security problem. As whole user data is stored in the cloud, the security is the definitely without doubt is one of the main issues in cloud.

*Bandwidth:* Security concerns have long dominated much of the cloud conversation and caused many companies to deliberate about getting started in the cloud. But while the focus has been on cloud security, another potential bottlenecks are on the way like bandwidth requirement. Since bandwidth is rarely a problem for companies exploring the cloud in a small way. But as they start expanding their cloud footprint and running production-oriented applications, data movement takes on a completely different scale. As enterprises start to move real workloads out to the cloud look for bandwidth to become top of mind. The data turned to be big one for the enterprise so that they directly depend on the bandwidth problem.

*Energy Consumption:* Cloud computing is rapidly growing in importance as increasing numbers of enterprises and individuals are shifting their workloads to cloud service providers. Services offered by cloud providers such as Amazon, Microsoft, IBM, and Google are implemented on thousands of servers spread across multiple geographically distributed data centers. The electricity costs involved in operating a large cloud infrastructure of multiple data centers can be enormous. In fact, cloud service providers often must pay for the peak power they draw, as well as the energy they consume. Lowering these high operating costs is one of the challenges facing cloud service providers. Moreover, there are other crucial problems that arise from high power consumption. Insufficient or malfunctioning cooling system can lead to overheating of the resources reducing system reliability and devices lifetime. In addition, high power consumption by the infrastructure leads to substantial carbon dioxide (CO<sub>2</sub>) emissions contributing to the greenhouse effect. Geographical centers exposes many opportunities for cost savings due to more energy consumption. The data centers are often exposed to different electricity markets, meaning that they pay different energy and peak power prices. Finally, the data centers may be located in areas with widely different outside temperatures, which have an impact on the amount of cooling energy used. Anyhow the cloud computing affects the environment.

*Disaster recovery and business continuity:* Disaster recovery is one of the main issue in cloud computing. If a client stores his data over the cloud provided by the cloud service provider, suppose the cloud service provider suffered by the sudden business loss and the cloud service provider is in a position that made to the shutdown of the data server or bankrupted by the agency. Then the client data are in an uncertain state. It can't be recovered by the client and the cloud service provider is also unable to answer for the client on this issues, this is the disaster recovery and the business continuity problem in the cloud computing.

*Availability:* The availability issues in cloud computing is similar to the disaster recovery and the business continuity in cloud computing. The data server present in the cloud service provider must provide uninterrupted service to the client side in order to avoid down time in the client side. In the cloud, downtime or data loss can quickly cripple workflow and impose substantial costs from time and money spent on repairs and lapses in productivity, to the inability to meet service levels.

## **1.2 SECURITY ISSUES IN CLOUD**

When it comes to Security, cloud really suffers a lot. The vendor for Cloud must make sure that the customer does not face any problem such as loss of data or data theft. There is also a possibility where a malicious user can penetrate the cloud by impersonating a legitimate user, there by infecting the entire cloud thus affecting many customers who are sharing the infected cloud. Some of the problem which is faced by the Cloud computing.

### **1.2.1 NEED FOR SECURITY IN CLOUD COMPUTING**

The security issues is the most important issues in cloud computing. The cloud computing has the greatest advantage of becoming a must to all enterprise , to make it possible some of the security issues in cloud computing has to be rectified and solution to be provided for all the security related problem. If security issues are not solved then cloud computing can't sustain in the computing industry. To make the cloud computing to accept in all industry then security issues needs to be nullified. Both the cloud user and cloud service provider must make sure that the data are safe without any external or internal threat.

### 1.2.2 DIFFERENT SECURITY ISSUES IN CLOUD COMPUTING

There are various security issues that are present in cloud computing, some of the main issues are listed below

*Data Integrity:* When a data is on a cloud anyone from any location can access those data's from the cloud. Cloud does not differentiate between a sensitive data from a common data thus enabling anyone to access those sensitive data's. Thus there is a lack of data integrity in cloud computing.

*Data Theft:* Most of the cloud Vendors instead of acquiring a server tries to lease a server from other service providers because they are cost affective and flexible for operation. The customer doesn't know about those things, there is a high possibility that the data can be stolen from the external server by a malicious user.

*Privacy issues:* The Vendor must make sure that the Customer information or data is well secured from other operators. As most of the servers are external, the vendor should make sure who is accessing the data and who is maintaining the server thus enabling the vendor to protect the customer's information. Hacker should not damage the data of the user.

*Infected Application:* Vendor should have the complete access to the server for monitoring and maintenance, thus preventing any malicious user from uploading any infected application onto the Cloud and should not affect any user's data on the internet.

*Data loss:* Data loss is a very serious problem in Cloud computing. If the vendor closes due to financial or legal problems there will be a loss of data for the customers. The customers won't be able to access those data's because data is no more available for the customer as the vendor shut down.

*Data Location:* When it comes to location of the data nothing is transparent even the customer doesn't know where his own data's are located. The Vendor does not reveal where all the data's are stored. The Data's won't even be in the same country of the Customer, it might be located anywhere in the world.

*Security on Vendor level:* Vendor should make sure that the server is well secured from all the

external threats it may come across. A Cloud is good only when there is a good security provided by the vendor to the customers.

*Security on user level:* Even though the vendor has provided a good security layer for the customer, the customer should make sure that because of its own action, there shouldn't be any loss of data or damage of data for other users who are using the same Cloud.

*Data Confidentiality:* Data confidentiality is important for users to store their private or confidential data in the cloud. Authentication and access control strategies are used to ensure data confidentiality. The data confidentiality, authentication, and access control issues in cloud computing could be addressed by increasing the cloud reliability and trustworthiness by the cloud service provider.

### 1.2.3 THIRD PARTY BASED SECURITY MODEL

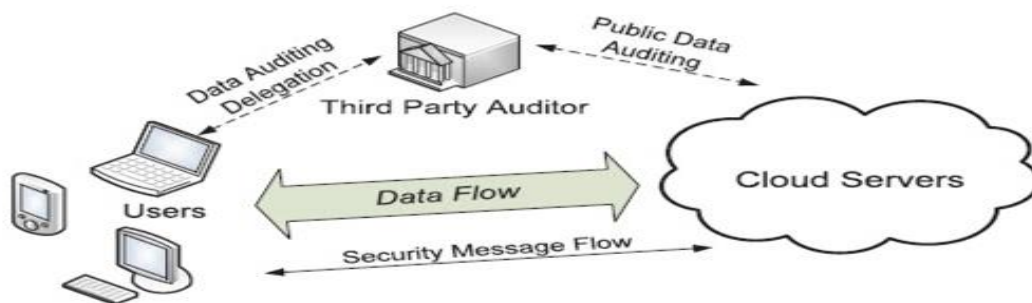


Fig 1.3 Cloud computing with trusted third party[3]

Third party allows user to simplify their task, third party does the work on behalf of the user and thereby reducing the burden on the user side. Third party are used for various purpose, one of the main purpose is for checking data integrity that is the data sent to the cloud service provider must be same as the data that is received by the user from the cloud service provider. Other purpose the third party is used were encryption and decryption purpose, authentication purpose and other little task that are needed for security aspects.



#### **1.2.4 NEED FOR SECURITY ENHANCEMENT IN CLOUD**

Though cloud computing is one of the best technology, to make it acceptable in large scale Security must never be compromised. The security issues is the most important issues in cloud computing. A Secure cloud is always a reliable source of information, thus protecting the cloud in a very important task for security professionals who are in charge of the cloud. Some of the ways by which a cloud can be protected are Protection of data, making sure data is available for the customers, delivering high performance for the Customers. And also to make sure the application used by the customer is safe to use, Most important of them all is that, there should be a good degree of encryption provided by the vendor to the customer that only the customer should be able to access the data and not the malicious User. Both the Vendor and the customer should make sure that the cloud is safe from all the external threats, thus there will be a mutual understanding between the customer and the vendor when it comes to the security on Cloud.

## **CHAPTER 2**

### **LITERATURE SURVEY**

#### **2.1 INTRODUCTION**

The fundamental factor defining the success of any new computing technology is the level of security it provides. Whether the data residing in the cloud is secure to a level so as to avoid any sort of security breach or is it more secure to store the data away from cloud in our own personal computers or hard drives. At-least we can access our hard drives and systems whenever we wish to, but cloud servers could potentially reside anywhere in the world and any sort of internet breakdown can deny us access to the data stored in the cloud. The cloud service providers insist that their servers and the data stored in them is sufficiently protected from any sort of invasion and theft. Such companies argue that the data on their servers is inherently more secure than data residing on a myriad of personal computers and laptops.

However, it is also a part of cloud architecture, that the client data will be distributed over these individual computers regardless of where the base repository of data is ultimately located. There have been instances when their security has been invaded and the whole system has been down for hours. At-least half a dozen of security breaches occurred last year bringing out the fundamental limitations of the security model of major Cloud Service Providers (CSP) .

Data confidentiality occurs because the users do not trust the cloud providers and cloud storage service providers are virtually impossible to eliminate potential insider threat, it is very dangerous for users to store their sensitive data in cloud storage directly. Simple encryption is faced with the key management problem and cannot support complex requirements such as query, parallel modification, and fine-grained authorization. The various problem areas in data confidentiality in cloud computing is listed and the works of various paper that deals with the data confidentiality is also discussed.

Encryption is usually used to ensure the confidentiality of data. Homomorphic encryption is a kind of encryption system proposed by Rivest et al. It ensures that the cipher text algebraic operation results are consistent with the clear operation after encryption results; besides, the

whole process does not need to decrypt the data. The implementation of this technique could well solve the confidentiality of data and data operations in the cloud.

Gentry firstly proposed the fully homomorphic encryption method [1], which can do any operation that can be performed in clear text without decrypting. It is an important breakthrough in the homomorphic encryption technology. However, the encryption system involves very complicated calculation, and the cost of computing and storage is very high. This leads to the fact that the fully homomorphic encryption is still far from real applications.

A cryptographic algorithm named Diffie-Hellman is proposed for secure communication [2], which is quite dissimilar to the key distribution management mechanism. For more flexibility and enhanced security, a hybrid technique that combines multiple encryption algorithms such as RSA, 3DES, and random number generator has been proposed [3]. RSA is useful for establishing secure communication connection through digital signature based authentication while 3DES is particularly useful for encryption of block data.

## **2.2 SEGREGATE SECURITY MODEL BASED ON TRUSTED THIRD PARTY**

There are numerous method that are proposed with the trusted third party in cloud computing and much of them are focused on providing security to cloud computing. Though cloud computing is a secure one, there are several chance that the cloud service provider may be untrusted as it has the chance to peek into the user data. This affects the privacy of the user. A third party audit to ensure the correctness and integrity of the data for multiple users simultaneously and efficiency with RC 5 encryption algorithm [5].

This method provides high efficient and low computation cost due to RC5 encryption and multiple users can be handled. Another model proposes a way to implement TPA which not only check the reliability of the cloud service provider, but also check the consistency and integrity of data using RSA algorithm on the client side. It includes digital signature method for the verifying the correctness of the data [6]. This model supports all the features, easy to implement and RSA is used and it provides more security to the user. There is model which says this paper proposes that User authentication can be done using TTP by assuming it is be trusted. It uses Salt and Hash function for the data integrity. The advantage is Login can be secured and it provides simple and effective method for authentication [7].

There is a specific method which successfully checks the data integrity of the data that is stored in the system [8]. This model proposed to address data confidentiality, data integrity and authentication with the help of TTP and it helps to check the correctness of the data. The advantages of this model is hashing is done in the Trusted Third Party side and algorithm used are RSA, DES, AES.

There are also some methods which use Merkle Hash Tree algorithm for ensuring the data integrity in the cloud computing [9]. The main objective is to provide data integrity using Merkle Hash Tree algorithm which is done with the help of TTP based on the request from the cloud user. The two main advantages of this model is Integrity of data can be easily monitored and it is more effective for data inspection. In Data Security Model for Cloud Computing, from the perspective of data security, which has always been an important aspect of quality of service, Cloud computing focuses on new challenging security threats. Therefore, a data security model must solve the most challenges of cloud computing security.

The proposed data security model provides a single default gateway as a platform. It is used to secure sensitive user data across multiple public and private cloud applications, including Salesforce, Chatter, Gmail, and Amazon Web Services, without influencing functionality or performance. Default gateway platform encrypts sensitive data automatically in real time before sending to the cloud storage without breaking cloud application. It did not effect on user functionality and visibility.

If an unauthorized person gets data from cloud storage, he only sees encrypted data. If authorized person accesses successfully in his cloud, the data is decrypted in real time for your use. The default gateway platform must contain strong and fast encryption algorithm, file integrity, malware detection, firewall, tokenization and more. This paper interested about authentication, stronger and faster encryption algorithm, and file integrity.

In Hybrid Technique, a hybrid technique is proposed for data confidentiality and integrity, which uses both key sharing and authentication techniques. The connectivity between the user and the cloud service provider can be made more secure by utilizing powerful key sharing and authentication processes.

RSA public key algorithm can be used for secure distribution of the keys between the user and cloud service providers. A three-layered data security technique is proposed, the first layer is used for authenticity of the cloud user either by one factor or by two factor

authentications; the second layer encrypts the user's data for ensuring protection and privacy, and the third layer does fast recovery of data through a speedy decryption process.

Data confidentiality occurs because the users do not trust the cloud providers and cloud storage service providers are virtually impossible to eliminate potential insider threat, it is very dangerous for users to store their sensitive data in cloud storage directly. Simple encryption is faced with the key management problem and cannot support complex requirements such as query, parallel modification, and fine-grained authorization. The various problem areas in data confidentiality in cloud computing is listed and the works of various paper that deals with the data confidentiality is also discussed.

## **CHAPTER 3**

### **PROBLEM DEFINITION**

#### **3.1 EXISTING SOLUTIONS**

There are several disadvantages in using the existing solutions. Some of them are trusted third party concentrates more on data inspection rather than data securing in the cloud, it don't involve the privacy protection of the data, it just do the process of checking data integrity in the system and it creates more burden on the client side by encryption and decryption the data in the client side so that load on the client side and it leads to several fatal disadvantage. The existing solve the problem of data confidentiality by using certain kind of encryption and decryption in the user side and it solves the data confidentiality to some extend but it badly affects the performance of the system.

In order to overcome the problem of using encryption and decryption in the user side, few proposed some work uses the trusted third party in the middle of the user and the cloud service provider. The process of encryption and decryption occurs in the trusted third party but only few algorithm were used and the keys has to be stored in the user side, it has become a complex task as it makes inconvenience to the user. Another method uses the trusted third party for the purpose of checking whether the saved data is same as the data that is retrieved by the user data. It is used for the purpose of data integrity alone.

For the purpose of authentication, the trusted third party comes into existence; this method uses some kind of one time password for the purpose of authentication in the system. There are several method that is proposed for reducing the security issues. But most of the proposed method concentrates on data verification and inspection, those method does not concentrate more data securing in the system.

#### **3.2 ISSUES IN THE EXISTING SYSTEM**

One of the main drawbacks of cloud is that there are too much of possibility for the cloud service provider for the misuse of the data that is stored in their data center by the client. Due to this, whatever methods that are proposed don't have direct impact to reduce this problem. There

will be issues continuing in the cloud computing until the cloud service provider knowledge about the data is prohibited. There is also repeated usage of OTP method in the cloud computing techniques which makes this system to an inefficient one.

One of the main issues in existing methods are data verification is not concentrated more in the system. The data integrity is the key part of existing solutions with the help of trusted third party auditor in the system. And also the process of data encryption and decryption occurs in the cloud user side, so it increases burden for the user. No effective authentication method is used. There is no choice of encryption techniques for user. Key is stored in the user side, so it is uneasy for the user to maintain the key and TTP is not effectively used by the various methods that are provided by the various models.

There are several disadvantages in using the existing methods. Some of them are trusted third party concentrates more on data inspection rather than data securing in the cloud, it don't involve the privacy protection of the data and it has more burden on the client side as encryption and decryption taking place in the client side so that load on the client side will increase and makes the client to overload as all the process are carried out by the system.

## CHAPTER 4

### PROPOSED WORK

#### 4.1 PROPOSED ARCHITECTURE

The cloud computing has major setback of mistrust of the cloud service provider by the user. To overcome the problem we proposing a model with trusted third party. It is assumed here that the trusted third party is secured one. The trusted third party takes care of the major factor of security called as data confidentiality. The process of data confidentiality is satisfied by encrypting and decrypting the data with the user preference on that trusted third party side on behalf of the user. This process will reduce the overall workload of the user. This process of encrypting the data in the cloud service provider reduces the overall work of the user and the user is provided with various algorithms that increase the user experience and thus preventing the privacy of the user in the cloud environment.

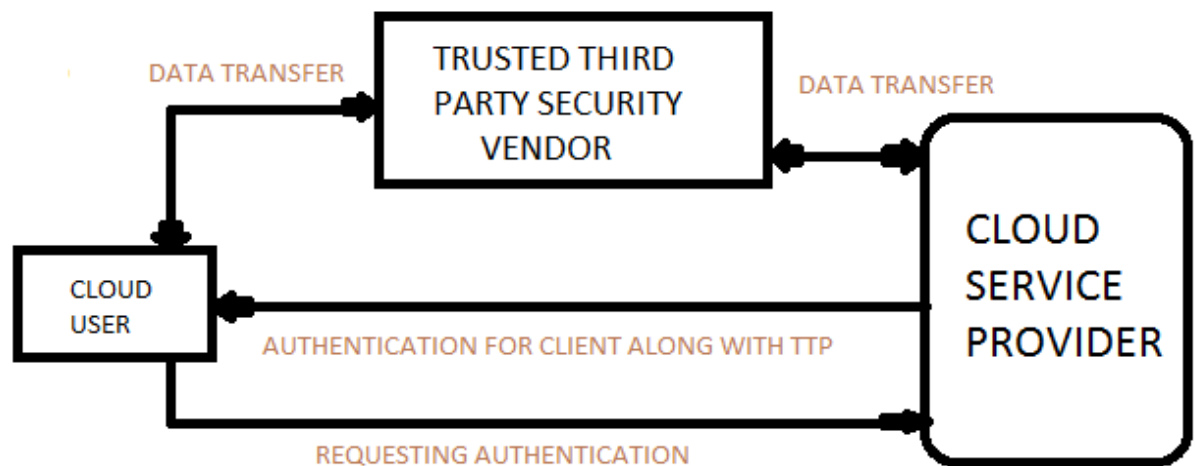


Fig 4.1 Proposed models with trusted third party



- A) The cloud user logs in to the cloud service provider by their required login id and the password.
- B) The user store and retrieve the data traditionally as in usual system.
- C) If the user wants to store some highly confidential data, the user goes to the cloud service provider by clicking on the TTP icon.
- D) In the TTP, the user can encrypt the required text file and it is stored in the homepage of the user in the cloud service provider.
- E) If the user wants to decrypt the encrypted file, the user can again go for the TTP and decrypt the file.
- F) The decrypted content will be displayed in the homepage of the user in the cloud service provider.

## **4.2 MODULES DESCRIPTION**

*Cloud User (CU):* User is an entity, which has large data files to be stored in the cloud and relies on the cloud for data storage, can be either individual consumers or organizations. Also it is totally responsible for storage data.

*Trusted Third Party (TTP):* TTP is an entity, which has expertise and capabilities for Encryption and decryption Service. When client want to store data at the cloud storage at that time (encryption/decryption service) Encrypt the data and stores in the cloud storage provider.

*Cloud Storage Provider (CSP):* CSP is an entity which is totally responsible for storage of the data. After encrypting the data the TTP stores the data on cloud storage provider. It provides the connection between the CSP and TTP.

### 4.3 ALGORITHM

To encrypt data at cloud storage both symmetric-key and asymmetric-key algorithms can be used [8]. Cloud storage contains a large set of databases and for such a large database asymmetric-key algorithm's performance is slower when compared to symmetric-key algorithms. Algorithms such as DES, RC4, 3DES, AES, Blowfish are used for encryption and decryption purpose.[9] The AES is the fastest symmetric technique as well as it can be implemented simply. After then, the symmetric techniques can be ordered as DES, and finally the Blowfish.

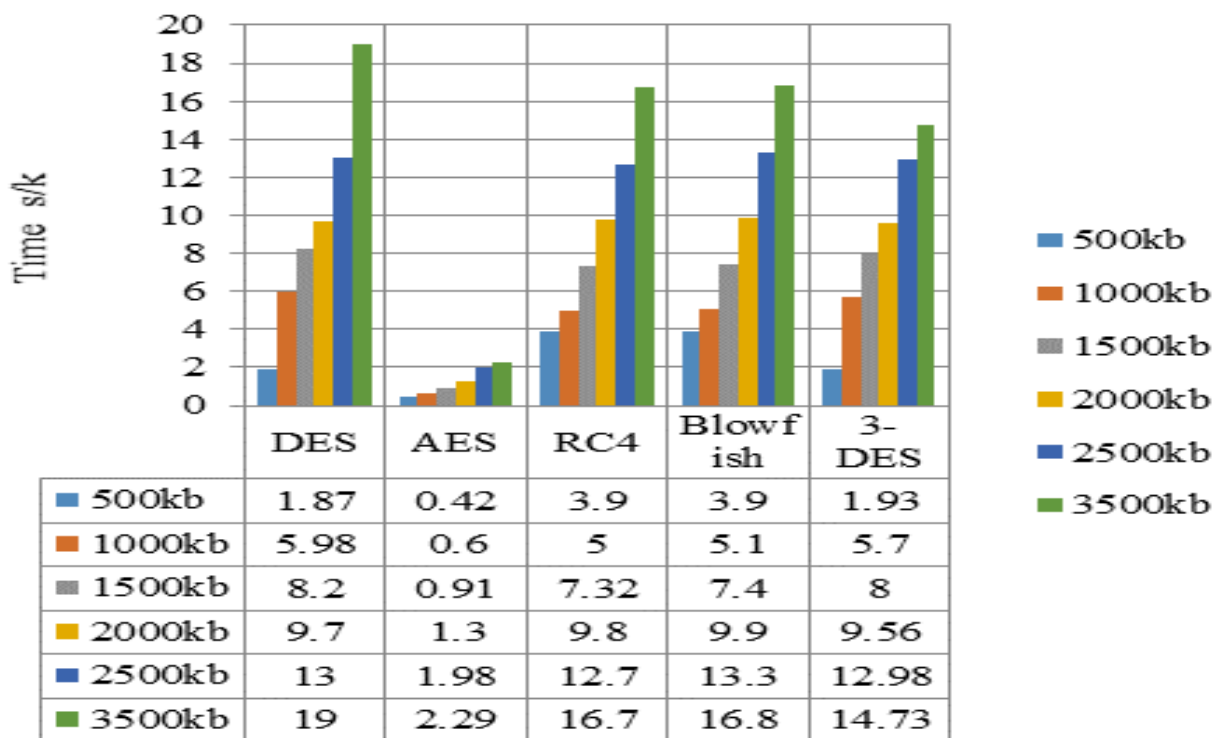


Fig 4.2 Algorithm efficiency comparison [11]

*DES Algorithm:* The Data Encryption Standard is a block cipher, meaning a cryptographic key and algorithm are applied to a block of data simultaneously rather than one bit at a time. To encrypt a plaintext message, DES groups it into 64-bit blocks.

*AES Algorithm:* The Advanced Encryption Standard (AES), also referenced as Rijndael (its original name), is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.

*Blowfish Algorithm:* Blowfish is a symmetric-key block cipher, designed in 1993 by Bruce Schneier and included in a large number of cipher suites and encryption products. Blowfish provides a good encryption rate in software and no effective cryptanalysis of it has been found to date.

*3DES Algorithm:* Triple DES is the common name for the Triple Data Encryption Algorithm which applies the Data Encryption Standard (DES) cipher algorithm three times to each data block. Triple DES provides a relatively simple method of increasing the key size of DES to protect against such attacks, without the need to design a completely new block cipher algorithm.

*RC4 Algorithm:* The RC4 Encryption Algorithm, developed by Ronald Rivest of RSA, is a shared key stream cipher algorithm requiring a secure exchange of a shared key. The symmetric key algorithm is used identically for encryption and decryption such that the data stream is simply XORed with the generated key sequence. The algorithm is serial as it requires successive exchanges of state entries based on the key sequence.

## **CHAPTER 5**

### **SYSTEM SPECIFICATION**

Cloud computing is the use of computer resources (h/w (or) s/w) that is delivered as a service over network (typically internet). To implement this project we need to have certain software and components. These are two type of requirement; they are hardware and software requirement. The hardware components are those which are used for giving input and getting the output and for storage also. The software components are those which is used for programming purpose, how to store the information. They have been listed below.

#### **5.1 HARDWARE REQUIREMENTS**

The below hardware are required for the execution of the proposed system.

System	:	Pentium IV 3.5 GHz
Hard Disk	:	100 GB
Monitor	:	14' Colour Monitor
Mouse	:	Optical Mouse
Ram	:	2 GB

#### **5.2 SOFTWARE REQUIREMENTS**

The below software are required for the execution of the proposed system.

Operating system	:	Windows 8
Coding Language	:	PHP, HTML
Data Base	:	MySQL
Webserver	:	XAMPP
IDE	:	Notepad ++
Cloud Setup	:	OwnCloud

### 5.3 OBJECT ORIENTED ANALYSIS

Object-oriented analysis is a popular technical approach to analyzing, designing an application, system, or business by applying the object-oriented paradigm and visual modeling throughout the development life cycles to foster better communication and product quality.

#### 5.3.1 USE CASE DIAGRAM

A use case diagram is a representation of a user's interaction with the system and depicting the specifications of a use case. As refer in the figure 5.1, proposed model has four main modules such as cloud user, registered user, cloud service provider and the trusted third party.

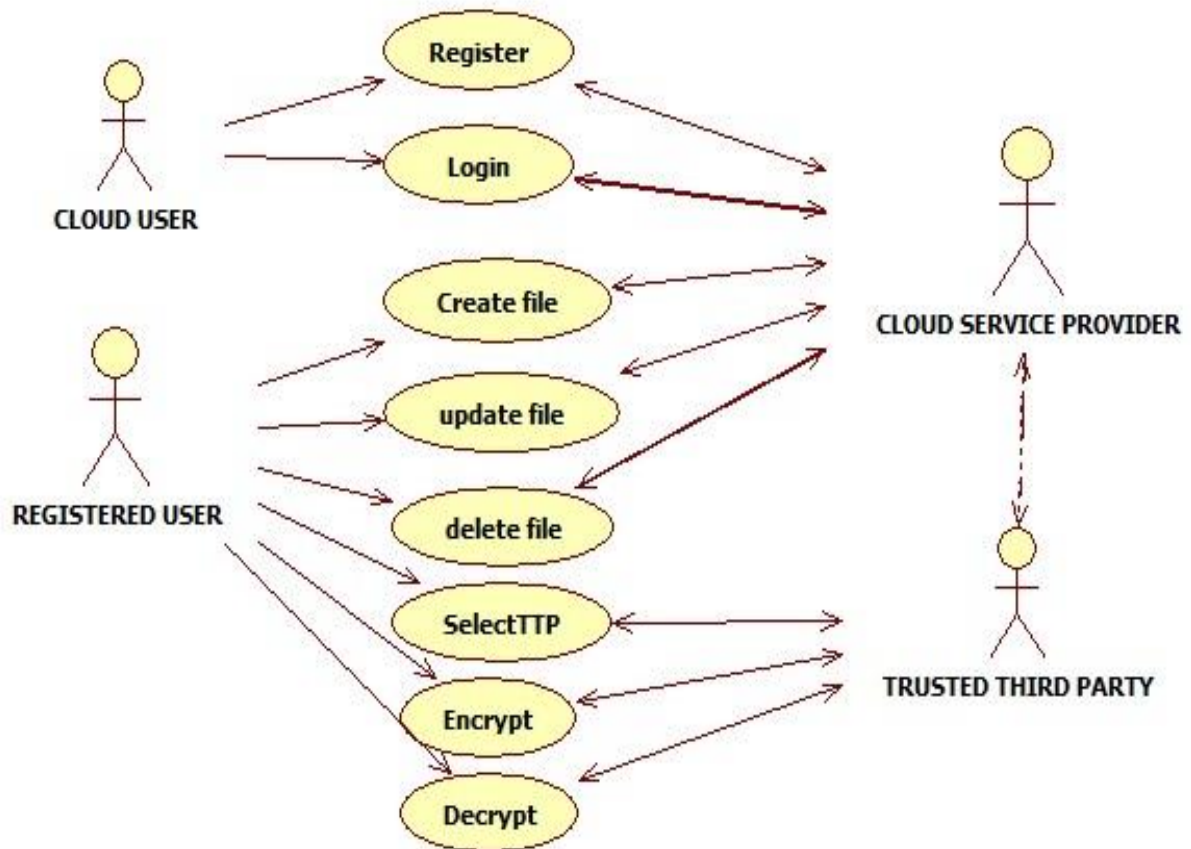


Fig 5.1 Use Case Diagram for the proposed system

### 5.3.2 CLASS DIAGRAM

The class diagram is the main building block of object oriented modelling. It is used both for general conceptual modelling of the systematics of the application, and for detailed modelling translating the models into programming code. As refer in the figure 5.2, there are three classes that are present in the system, the cloud provider has five functions and the TTP class has five methods that are used by the system. The user class contains login, register, upload, encrypt and decrypt functions.

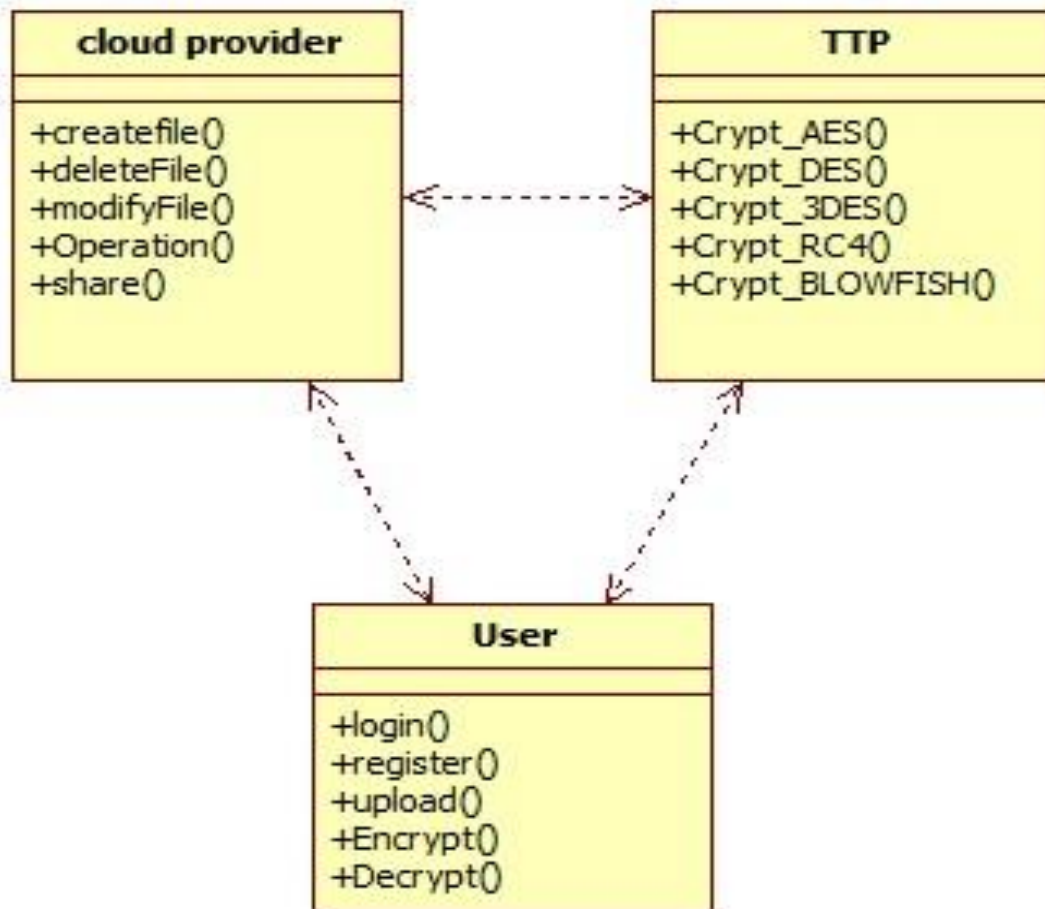


Fig 5.2 Class Diagram for the proposed system

### 5.3.3 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. The figure 5.3 depicts all the phases that are connected and flow of work between connected phases is shown in figure.

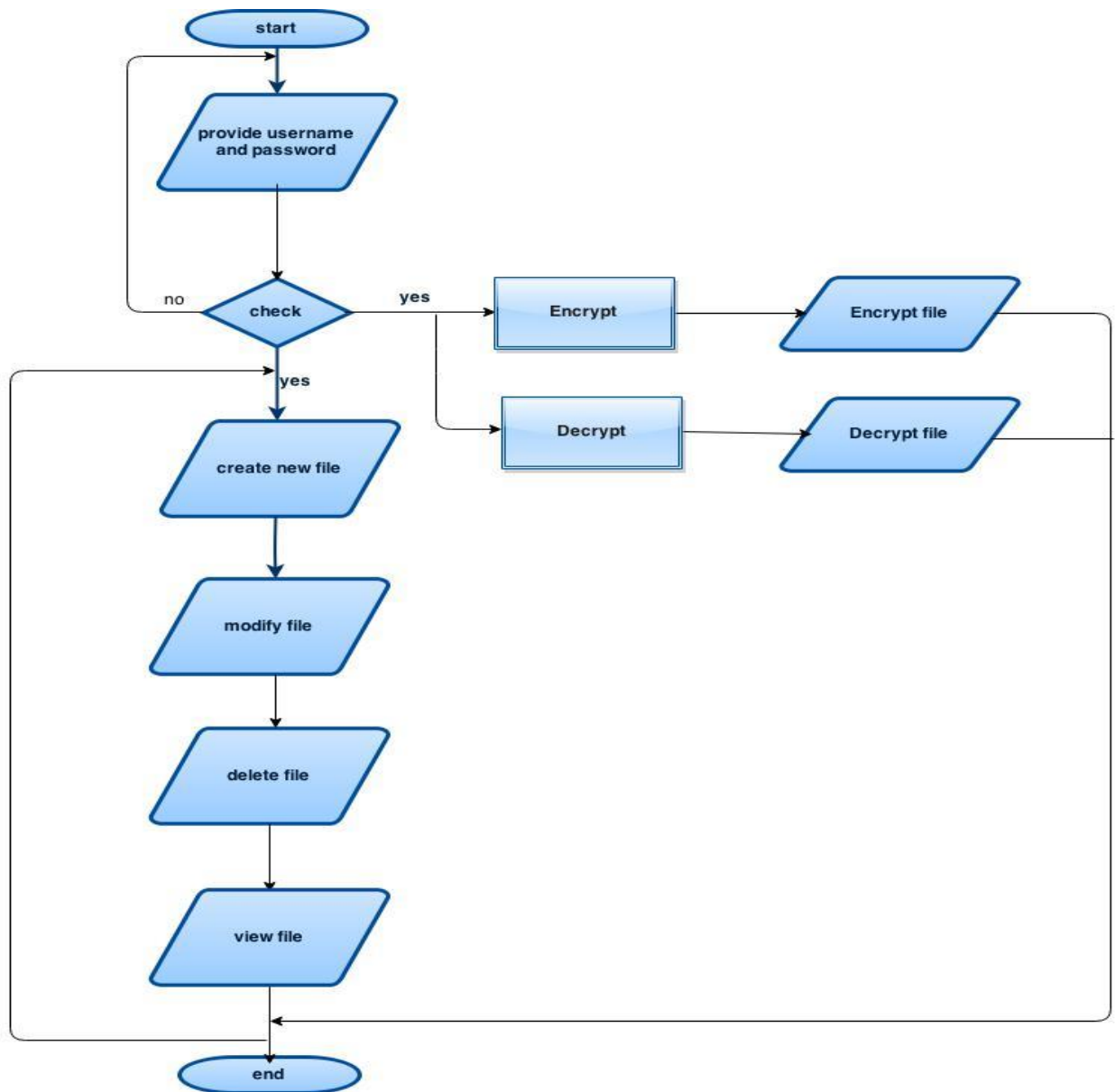


Fig 5.3 Activity Diagram for the proposed system

### 5.3.4 SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. The sequence of interaction between the cloud user, TTP and the cloud service provider is shown in the figure 5.4.

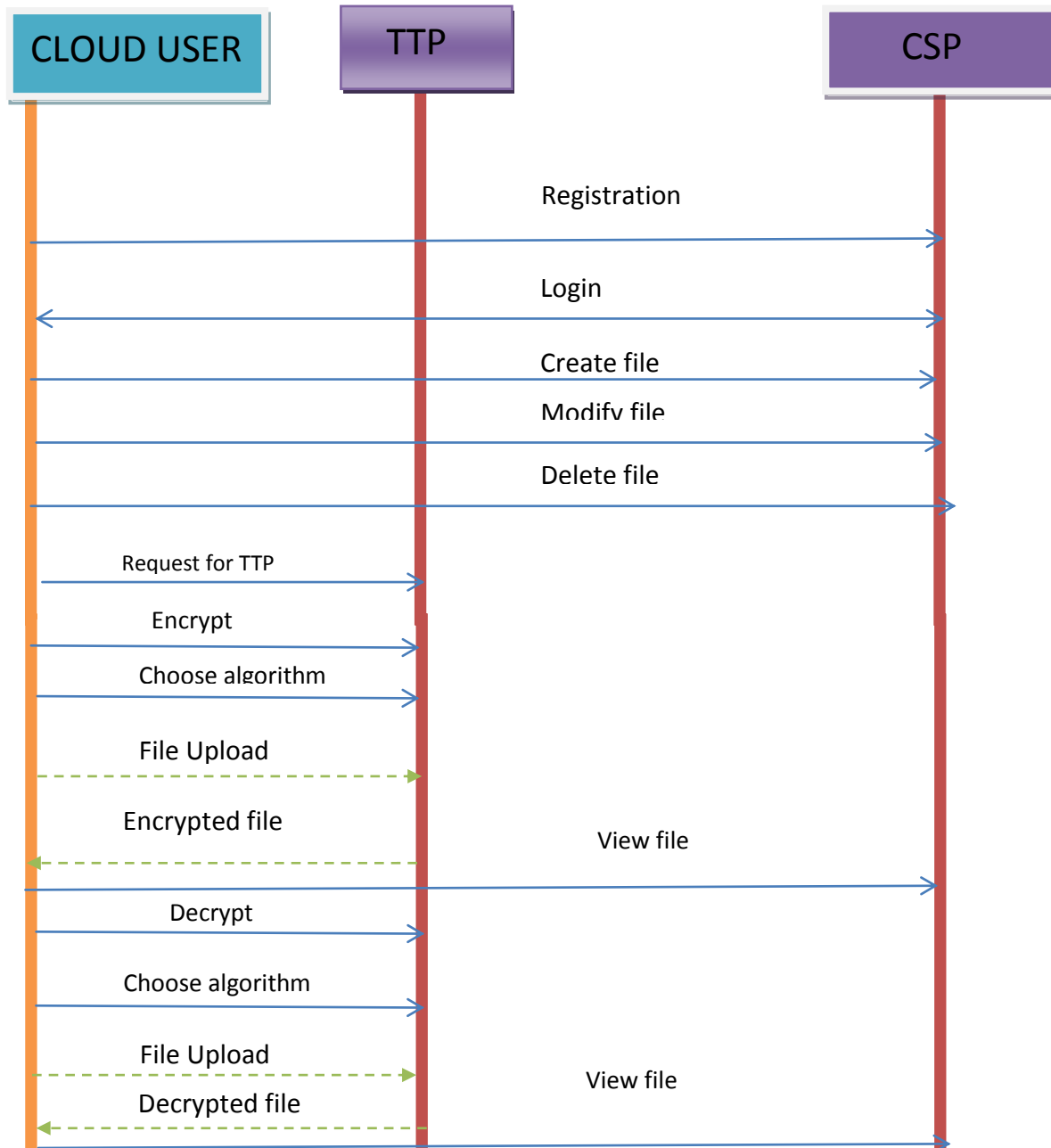


Fig 5.4 Sequence Diagram for the proposed system



## **CHAPTER – 6**

### **SYSTEM DESIGN AND IMPLEMENTATION**

#### **6.1 IMPLEMENTATION DETAIL**

There are two modules that are present in the system they are

- Cloud user module
- Trusted third party module

##### **6.1.1 CLOUD USER MODULE**

###### **6.1.1.1 LOGIN**

The user can login into system using login module. They use their user name and password to get authenticated into the cloud.

###### **6.1.1.2 FILE CREATION PHASE**

The user can create any number of text file in the cloud environment, by clicking on the new icon the user can create the test file and the user doubles click it to enter the content in the file and modify the file.

###### **6.1.1.3 FILE SHARING PHASE**

The user can share the created file with the other user in the cloud environment. The user has the options to share the file with password and the expiry date. The file is shared with the help of links.

###### **6.1.1.4 USING TRUSTED THIRD PARTY**

If the user feels that the data is confidential, the user can go for trusted third party. It is done by clicking the TTP icon in the upper left corner of the user dashboard. By clicking the icon the user gets redirected to the TTP website.

## **6.1.2 TRUSTED THIRD PARTY MODULE**

### **6.1.2.1 REGISTRATION**

This module is used for registering the new user and the user details will be stored in the database table “newmembers” in the “test” database. The registration includes receiving the name of the user, email, username and password of the user. After registering the user can login into the system using their user name and password.

### **6.1.2.2 LOGIN**

Registered user will be login to the system using login module. They use their user name and password registered during registration phase to enter into the cloud. Using this phase only a user can encrypt and decrypt their files and store in their ownCloud account page.

### **6.1.2.3 ENCRYPTION PHASE**

After the successful login, the TTP will ask for the encryption and decryption of the user files. In the encryption phase, the user has to upload the required file from the system and select the algorithm that suits the files. The user is provided with algorithm such as AES, DES, 3DES, RC4 and BLOWFISH.

### **6.1.2.4 DECRYPTION PHASE**

The user can decrypt the encrypted file anytime by selecting the same algorithm which is used during the encryption phase. The user has to upload the encrypted file to the TTP for the decryption. The decrypted file comes with the prefix decrypted in the file name.

## **6.2 EXPERIMENTAL SETUP**

The cloud environment is setup using the ownCloud software and the trusted third party is implemented in PHP programming language using the webserver XAMPP. The frontend is HTML and the database is MySQL.

*XAMPP*: XAMPP is an open-source web server package that works on various platforms. It is actually an acronym with X meaning “cross” platform, A for Apache HTTP server, M for MySQL, P for PHP, and P for Perl. XAMPP was designed to help webpage developers, programmers, and designers check and review their work using their computers even without connection to the web or internet. So, basically XAMPP may be used to stand as pages for the internet even without connection to it. It can also be used to create and configure with databases

written in MySQL and/or SQLite. And since XAMPP is designed as a cross-platform server package, it is available for a variety of operating systems and platforms like Microsoft Windows, Mac OS X, Linux, and Solaris. Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet.

To make this as easy as possible, many important security features are disabled by default. In practice, however, XAMPP is sometimes used to actually serve web pages on the World Wide Web a special tool is provided to password-protect the most important parts of the package. XAMPP also provides support for creating and manipulating databases in MySQL and SQLite among others. Once XAMPP is installed, it is possible to treat a local host like a remote host by connecting using an FTP client. Using a program like FileZilla has many advantages when installing a content management system (CMS) like Joomla or WordPress. It is also possible to connect to local host via FTP with an HTML editor. The default FTP user is "new user", the default FTP password is xampp. The default MySQL user is "root" while there is no default MySQL password

*OwnCloud:* OwnCloud is a software system for what is commonly termed "file hosting". As such, ownCloud is functionally very similar to the widely used Dropbox, with the primary functional difference being that ownCloud is free and open-source, and thereby allowing anyone to install and operate it without charge on a private server, with no limits on storage space or the number of connected clients. Despite the name, the software system does not use cloud computing unless Frank Karlitschek, a KDE software developer, started developing ownCloud in January 2010, in order to provide a free software replacement to proprietary storage service providers. OwnCloud is enterprise file sync and share that is hosted in your data center, on your servers, using your storage. OwnCloud provides Universal File Access through a single front-end to all of your disparate systems. Users can access company files on any device, anytime, from anywhere while IT can manage, control and audit file sharing activity to ensure security and compliance measures are met.

OwnCloud has been integrated with the GNOME desktop, Integration of ownCloud with the Kolab groupware and collaboration project has started as of 2013. Additional projects that use or link to ownCloud include a Raspberry Pi project to create a cloud storage system using the

Raspberry Pi's small, low-energy form-factor. it is manually configured. In order for desktop machines to synchronize files with their ownCloud server, desktop clients are available for PCs running Windows, OS X, FreeBSD or Linux. Mobile clients exist for iOS and Android devices. Files and other data (such as calendars, contacts or bookmarks) can also be accessed using a web browser without any additional software. Any updates to files are pushed between all computers or mobile devices connected to a user's account. The ownCloud server is written in the PHP and JavaScript scripting languages. For remote access, it employs SabreDAV, an open-source WebDAV server. OwnCloud is designed to work with several database management systems, including SQLite, Maria DB, MySQL, Oracle Database, and PostgreSQL.

*HTML:* Hypertext Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. It is written in the form of HTML elements consisting of tags enclosed in angle brackets. HTML tags most commonly come in pairs, although some tags represent empty elements and so are unpaired. The first tag in a pair is the start tag, and the second tag is the end tag. Web browsers can read HTML files and compose them into visible or audible web pages. Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages. HTML consists of a series of short codes typed into a text-file by the site author — these are the tags. The text is then saved as a html file, and viewed through a browser, like Internet Explorer or Netscape Navigator. This browser reads the file and translates the text into a visible form, hopefully rendering the page as the author had intended. Writing your own HTML entails using tags correctly to create your vision. You can use anything from a rudimentary text-editor to a powerful graphical editor to create HTML pages.

HyperText Markup Language (HTML) is used for creating and visually representing a webpage. HTML adds "markup" to standard English text. "Hyper Text" refers to links that

connect Web pages to one another, making the World Wide Web what it is today. By creating and uploading Web pages to the Internet, you become an active participant in the World Wide Web. HTML supports visual images and other media as well. HTML is the language that describes the structure and the semantic content of a web document. Content within a web page is tagged with HTML elements such as <img>, <title>, <p>, <div>, <picture>, and so forth. These elements form the building blocks of a website.

*PHP:* PHP Stands for "Hypertext Preprocessor." PHP is an HTML-embedded Web scripting language. This means PHP code can be inserted into the HTML of a Web page. When a PHP page is accessed, the PHP code is read or "parsed" by the server the page resides on. The output from the PHP functions on the page is typically returned as HTML code, which can be read by the browser. Because the PHP code is transformed into HTML before the page is loaded, users cannot view the PHP code on a page.

This make PHP pages secure enough to access databases and other secure information. PHP stands for PHP: Hypertext Preprocessor PHP is a widely-used, open source scripting language, PHP scripts are executed on the server, PHP is free to download and use and PHP is simple for beginners. PHP also offers many advanced features for professional programmers. PHP files can contain text, HTML, JavaScript code, and PHP code , PHP code are executed on the server, and the result is returned to the browser as plain HTML and PHP files have a default file extension of ".php". The characteristics of PHP are

- PHP can generate dynamic page content
- PHP can create, open, read, write, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, and modify data in your database
- PHP can restrict users to access some pages on your website
- PHP can encrypt data

## CHAPTER 7

### RESULTS AND DISCUSSIONS

Cloud computing resources must be compatible, high performance and powerful. High performance is one of the cloud advantages which must be satisfactory for each service. Higher performance of services and anything related to cloud have influence on users and service providers. Hence, performance evaluation for cloud providers and users is important. There are many methods for performance prediction and evaluation; we use the following methods in our evaluation are Evaluation based on criteria and characteristics and evaluation based on simulation. Another category which can be considered for evaluating cloud performance is classification of three layers of cloud services evaluation.

Nowadays , the term “performance” is more than a classic concept and includes more extensive concepts such as reliability, energy efficiency, scalability and soon. Due to the extent of cloud computing environments and the large number of enterprises and normal users who are using cloud environment, many factors can affect the performance of cloud computing and its resources. Some of the important factors considered in this paper are as follows,

*Security:* The impact of security on cloud performance may seem lightly strange, but the impact of security on network infrastructure has been proven. For example, DDoS attacks have wide impact on networks performance and if happen, it will greatly reduce networks performance and also be effective on response time too. Therefore, if this risk and any same risks threaten cloud environment, it will be a big concern for users and providers.

*Recovery:* When data in cloud face errors and failures or data are lost for any reason, the time required for data retrieval and volumes of data which are recoverable, will be effective on cloud performance. For example, if the data recovery takes a long time will be effective on cloud Performance and customer satisfaction, because most organizations are cloud users and have quick access to their data and their services are very important for them.

*Service level agreements:* When the user wants to use cloud services, an agreement will be signed between users and providers which describes user’s requests, the ability of providers, fees, fines etc. If we look at the performance from personal view, the better, more optimal and more timely the agreed requests, the higher the performance will be .This view also holds true for providers. Network bandwidth, this factor can be effective on performance and can be a

criterion for evaluations too. For example, if the bandwidth is too low to provide service to customers, performance will be low too.

*Storage capacity:* Physical memory can also be effective on the performance criteria. This factor will be more effective in evaluating the performance of cloud infrastructure [12].

*Buffer capacity:* If servers cannot serve a request, it will be buffered in a temporary memory. Therefore, buffer capacity effect on performance. If the buffer capacity is low, many requests will be rejected and therefore performance will be low.

*Disk capacity:* can also have a negative or positive impact on performance in cloud. Fault tolerance will have special effect on performance of cloud environment. As an example, if a data center is in deficient and is able to provide the minimum services, this can increase performance.

*Availability:* With easy access to cloud services and the services are always available, performance will be increase. Number of users, if a data center has a lot of users and this number is greater than that of the rated capacity, this will reduce performance of services.

*Location:* Data centers and their distance from a user's location are also an important factor that can be effective on performance from the users' view.

Research and evaluation of wide environments usually associated with simulation. Also cloud computing is no exception of this rule, because research on total context of internet is too difficult and involves interaction with multiple computing and network elements, which may not be under control of developers. Moreover, network conditions may not be controllable and predictable and this will affect evaluation

The Trust is viewed as a belief that utilizes experience, to make trustworthy decisions. It is originally used in social science in constructing human beings' relationship and is now an essential substitute for forming security mechanism in distributed computing environments, as trust has many soft security attributes, such as, reliability, dependability, confidence, honest, belief, trustfulness, security, competence, and suchlike. In fact, trust is the most complex relationship among entities because it is extremely subjective to analyze and it's the user who can rate a system whether it is trusty or non-trusty, so user feedback can be considered to rate.

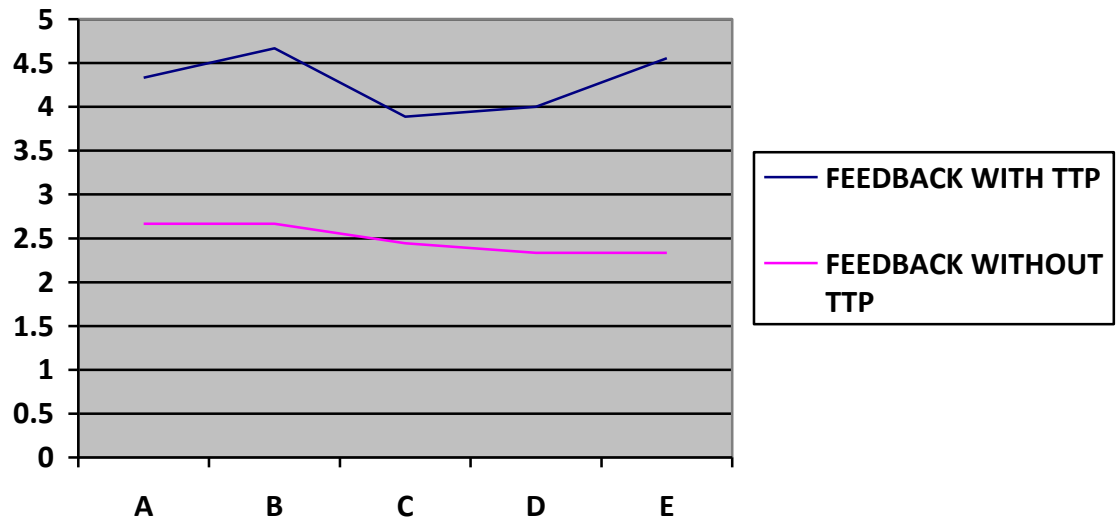


Fig 7.1 A graph for user feedback on trust [15]

	Nuppor et al	Mohta et al	Patel et al	Our work
Data confidentiality	✓	✓	✓	✓
Reduced workload	✗	✓	✓	✓
Multiple choice of encryption for user	✗	✗	✓	✓
Data Privacy	✗	✗	✗	✓

Table 7.1 Comparison of trust with the exiting work



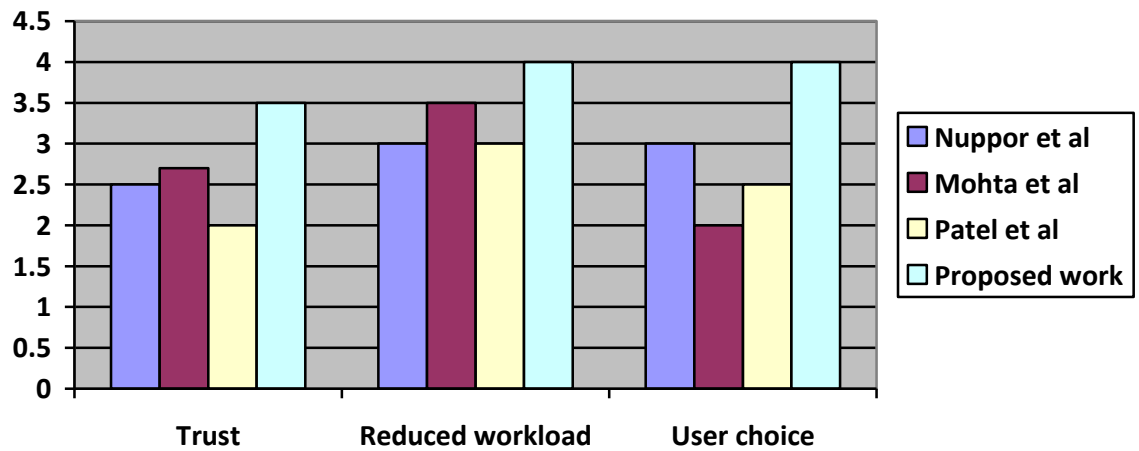


Fig 7.2 A graph based on various parameters

S.NO	Performance factor	Existing system			Proposed work
		Nuppor et al	Mohta et al	Patel et al	
1	Trust	2.5	2.7	2	3.5
2	Reduced Workload on client	3	3.5	3	4
3	User choice	3	2	2.5	4

Table 7.2 comparison of proposed work with existing methods

Based on the feedback received from the user, the values are computed for the various parameters that are involved in this proposed system. The parameters that are measured are trust, reduced workload on client and the user choice. From the evaluation, the proposed work has high value rather than the existing model that are proposed earlier by other methodologies.

## **CHAPTER 8**

### **CONCLUSION AND FUTURE ENHANCEMENT**

#### **8.1 CONCLUSION**

The above mentioned model is fruitful in getting the user to trust the cloud computing so that the more potential customer can use cloud computer without fear of misuse of data. The Encryption Algorithm applicability provides the flexibility in range and sequence to the user's choice because of the various Methods a user can apply based on their need and this prevent the cloud service provider from peeking into the user data. Thus preventing the privacy of the user and also makes the system more efficient. This method fulfills data confidentiality, thus achieving security for the user's data and provides more options for the user.

#### **8.2 FUTURE ENHANCEMENT**

The proposed system can be extended to solve the issues of data integrity and data authentication. The data integrity can be added by using some hash function in the encryption phase with the help of hash generator and in the decryption phase the hash verified can be used to check the data integrity in the cloud service provider. Authentication can be included by using the One Time Password system with the help of trusted third party and providing the data authentication in the system. Thus the system can be extended so that all three major issues such as data confidentiality, data integrity and data authentication can be solved.

## REFERENCES

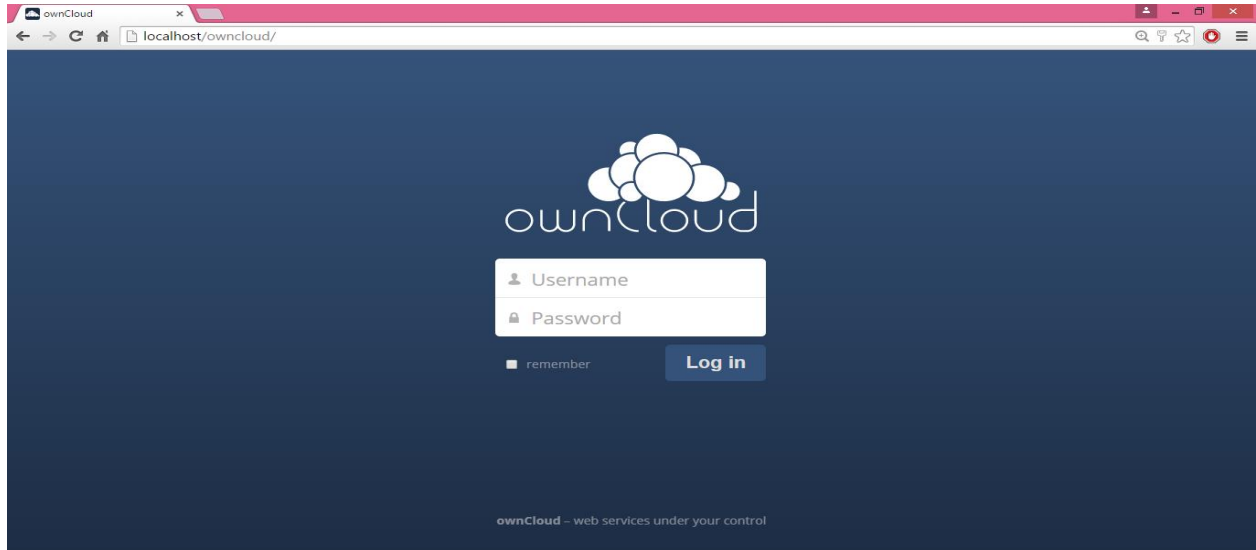
- [1]. Shuai Han, Jianchuan Xing, "Ensuring data security security through a novel third party auditor scheme in cloud computing", Proceedings of IEEE CCIS2011.
- [2]. [http://en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing)
- [3]. Figure Retrieved from [http://en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing)
- [4]. <http://thoughtsoncloud.com/2014/02/top-7-most-common-uses-of-cloud-computing/>
- [5]. Nuppor M.Yawale, V.B.Gadichcha," Third Party Auditing(TPA) for data storage security in cloud with RC5 algorithm" , International Journal of Advanced Research in Computer Science and Software Engineering 2013.
- [6]. Abhishek Mohta, Ravi Kant Sahu, Lalit Kumar Awasthi, " Robust Data Security for cloud while using Third party auditor", International Journal of Advanced Research in Computer Science and Software Engineering 2012.
- [7]. Susmita J A Nair,Anitha K L,Rosita F Kamala," Trusted Third party Authentication in cloud computing",International journal of Engineering Research and Technology 2013
- [8]. Patel Himani Atulkumar, Patel Srushti Hasmukhbhai," Cloud Model –With TPA (Third Party Auditor)", International journal of Research in engineering and advanced technology 2013.
- [9]. K.Meenakshi, Victo Sudha George," Cloud Server Storage Security using TPA", International Journal of Advanced Research in Computer Science and Technology"2014.
- [10] Reference for form : <http://goo.gl/forms/NADHHyxMGc>
- [11] Omer K. Jasim, Safia Abbas, El-Sayed M. El-Horbaty ,Abdel-Badeeh M. Salem," Efficiency of Modern Encryption Algorithms in Cloud Computing" in International Journal of Emerging Trends & Technology in Computer Science 2013.
- [12] Niloofar Khanghahi and Reza Ravanmehr," Cloud computing performance Evaluation: issues and challenges" in International Journal on Cloud Computing: Services and Architecture (IJCCSA) ,Vol.3, No.5, October 2013.

# APPENDICES

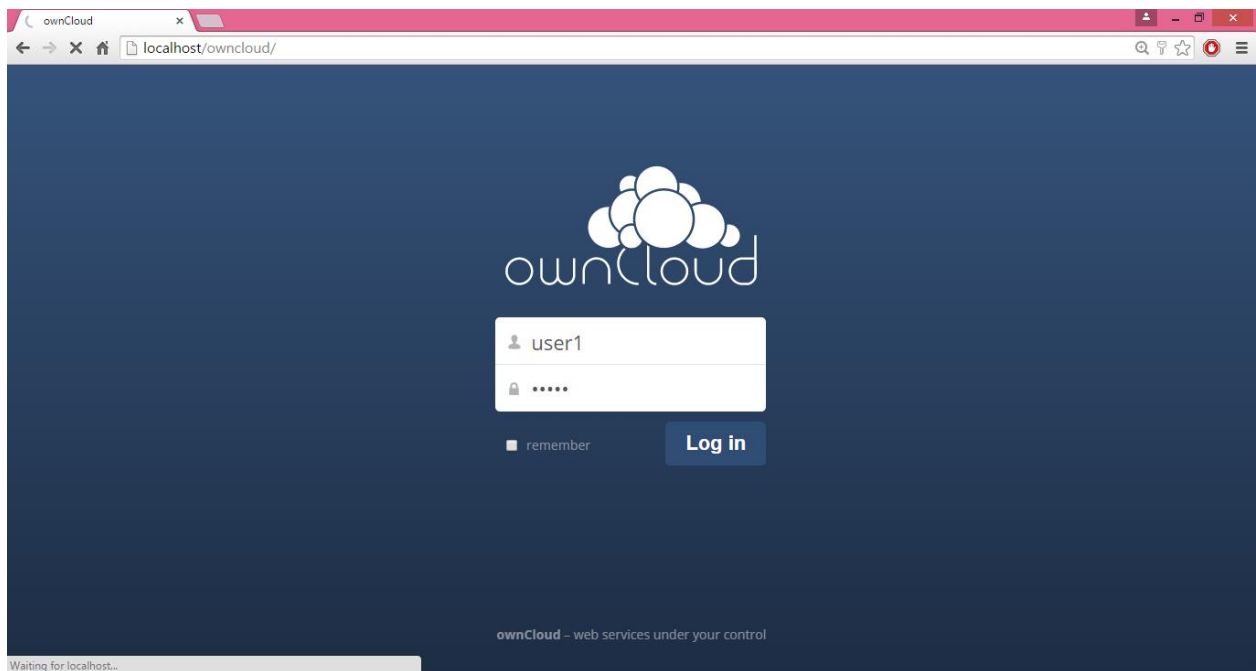
## APPENDIX 1

### SAMPLE SCREEN SHOTS

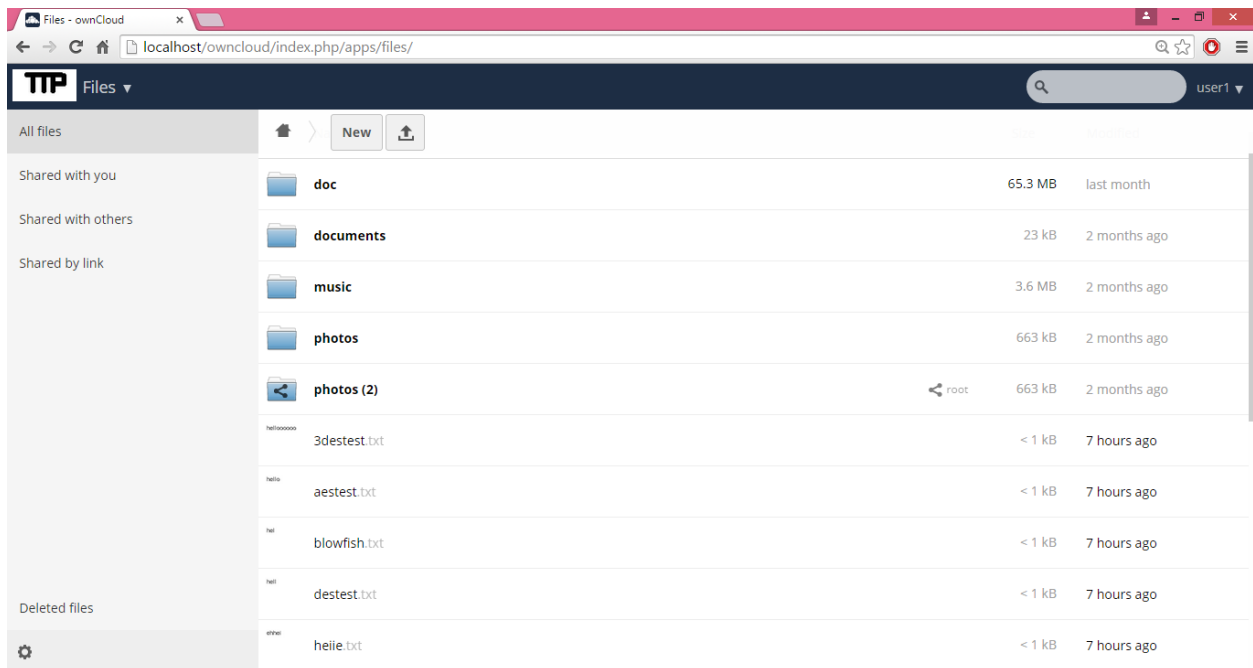
Login Screen:



User authentication:



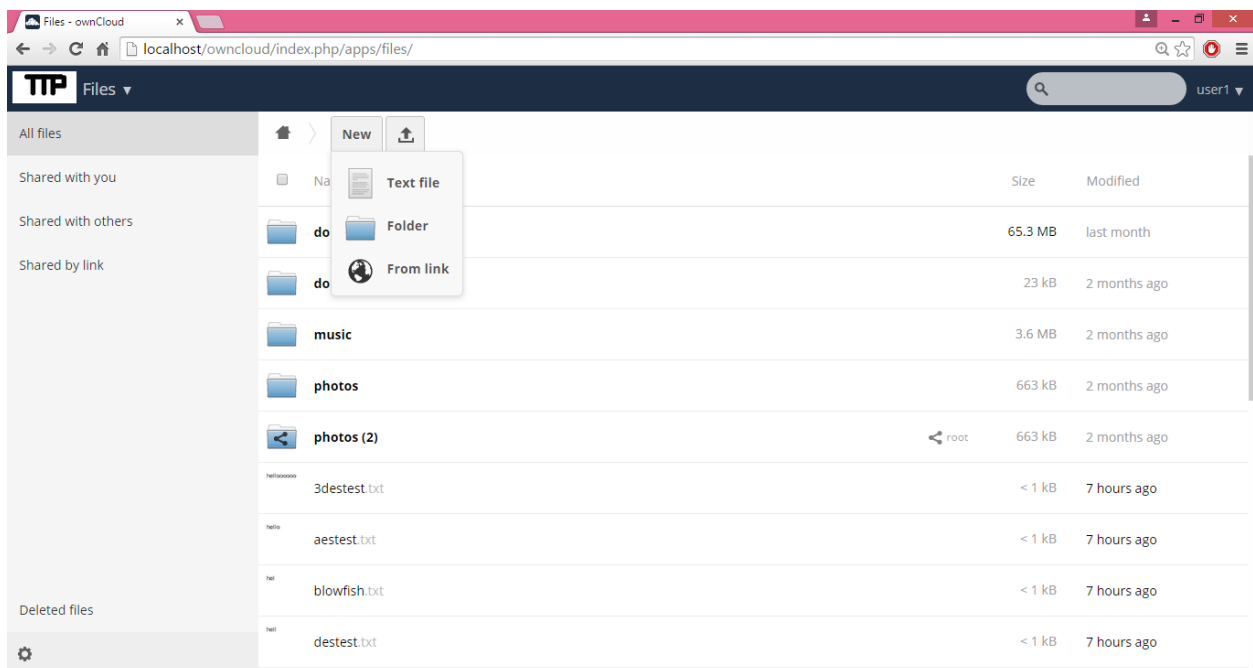
## User dashboard:



The screenshot shows the ownCloud user dashboard in a web browser. The address bar displays `localhost/owncloud/index.php/apps/files/`. The interface includes a sidebar on the left with navigation options: "All files", "Shared with you", "Shared with others", "Shared by link", and "Deleted files". The main area shows a file list with columns for "Name", "Size", and "Modified". The file list includes folders like "doc", "documents", "music", "photos", and "photos (2)", as well as text files like "3destest.txt", "aetestest.txt", "blowfish.txt", "destest.txt", and "heile.txt".

Name	Size	Modified
doc	65.3 MB	last month
documents	23 kB	2 months ago
music	3.6 MB	2 months ago
photos	663 kB	2 months ago
photos (2)	663 kB	2 months ago
3destest.txt	< 1 kB	7 hours ago
aetestest.txt	< 1 kB	7 hours ago
blowfish.txt	< 1 kB	7 hours ago
destest.txt	< 1 kB	7 hours ago
heile.txt	< 1 kB	7 hours ago

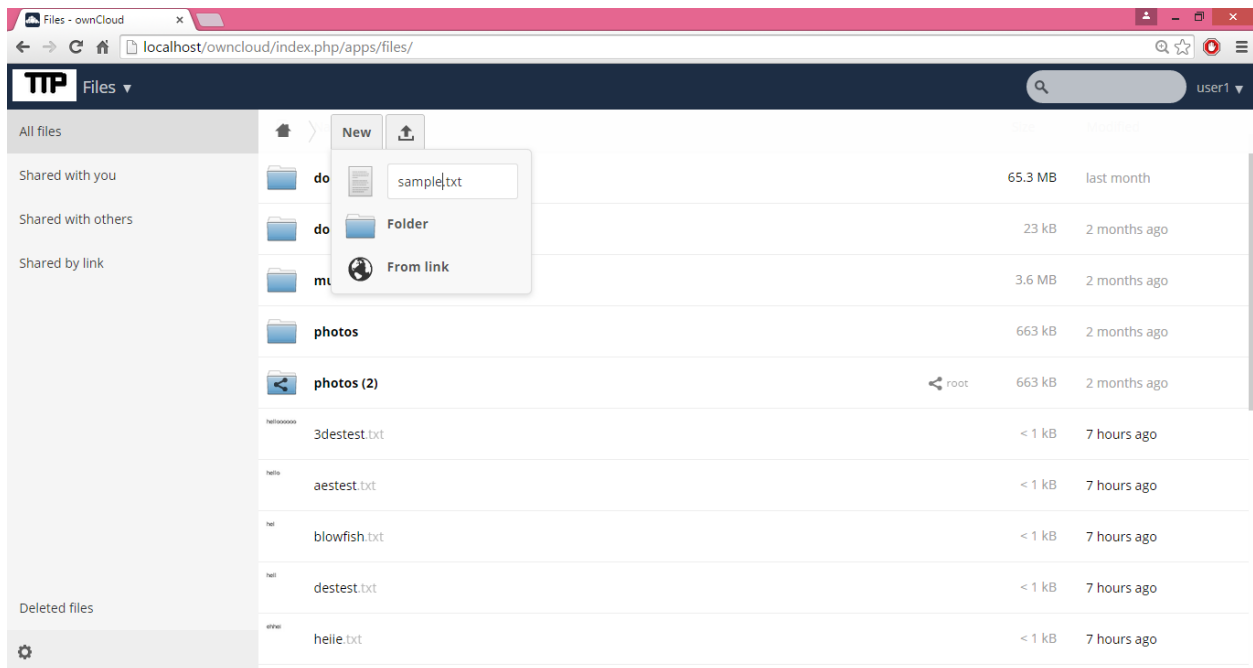
## New file creation:



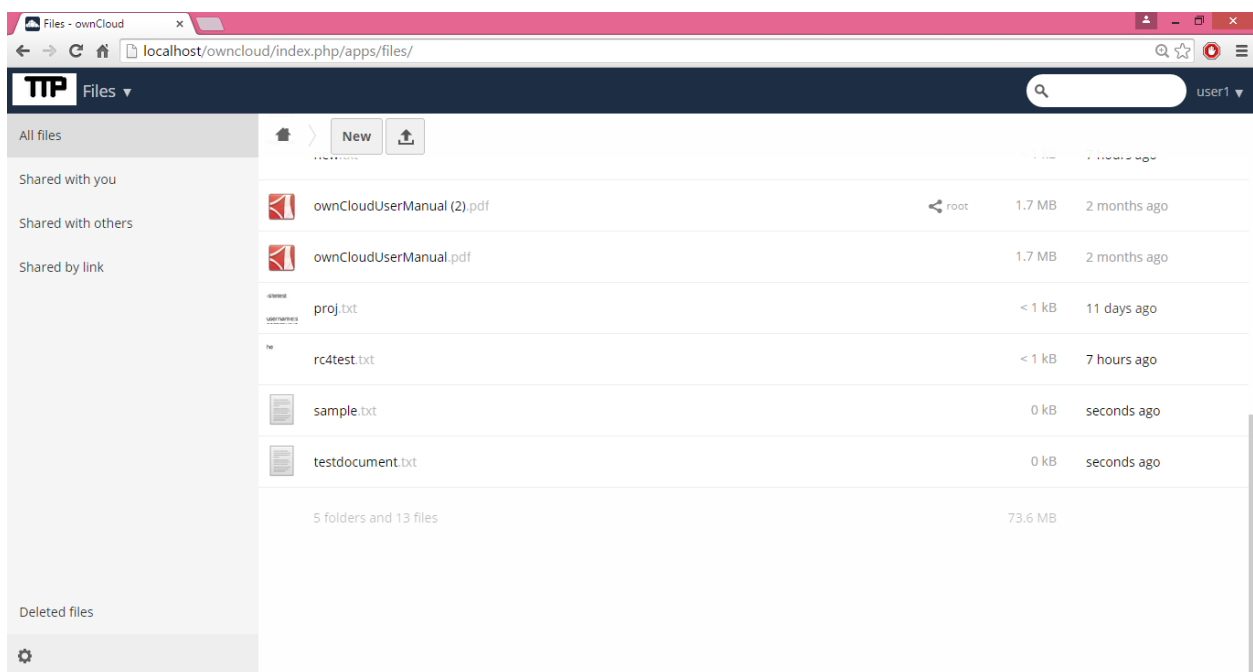
The screenshot shows the same ownCloud user dashboard, but with the "New" button in the top right corner of the file list area clicked. A dropdown menu is visible, offering three options: "Text file", "Folder", and "From link". The file list below the menu remains the same as in the previous screenshot.

Name	Size	Modified
doc	65.3 MB	last month
documents	23 kB	2 months ago
music	3.6 MB	2 months ago
photos	663 kB	2 months ago
photos (2)	663 kB	2 months ago
3destest.txt	< 1 kB	7 hours ago
aetestest.txt	< 1 kB	7 hours ago
blowfish.txt	< 1 kB	7 hours ago
destest.txt	< 1 kB	7 hours ago

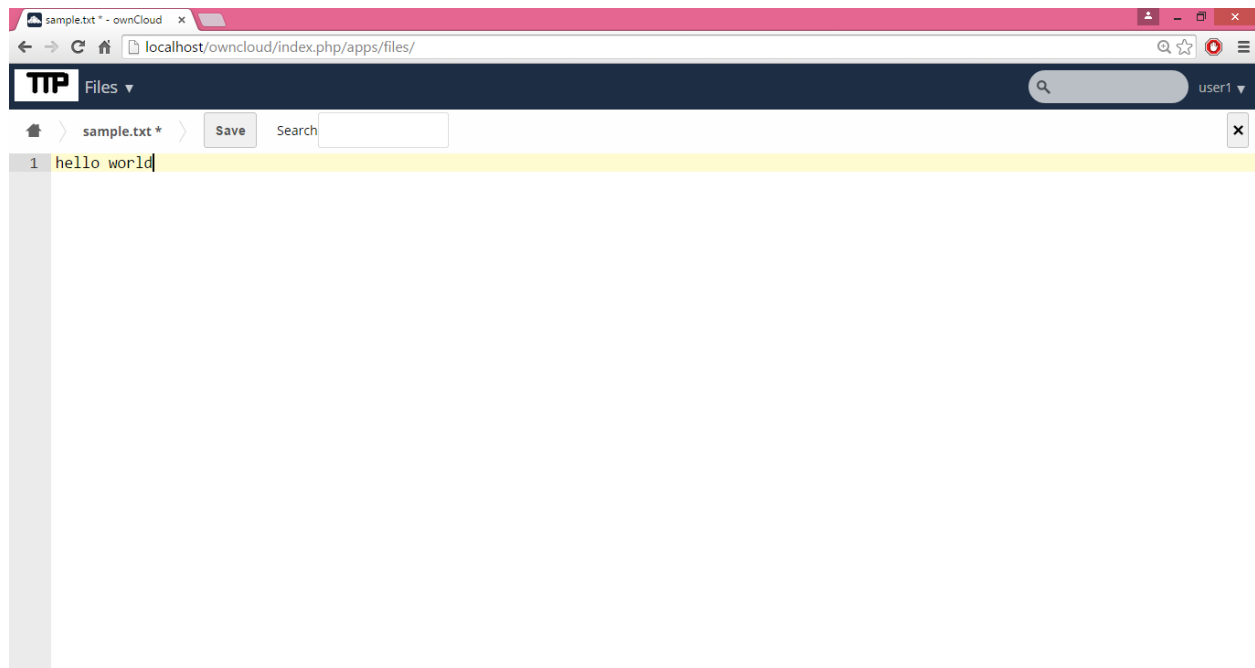
## Naming new file:



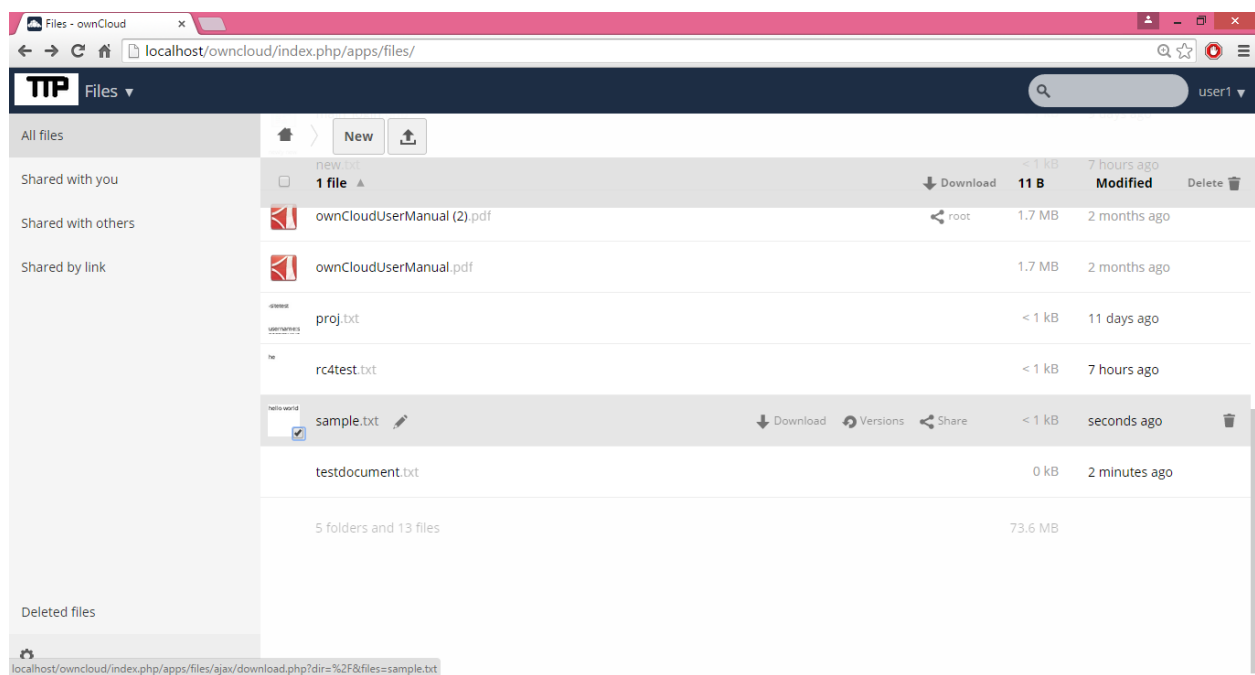
## New file in dashboard:



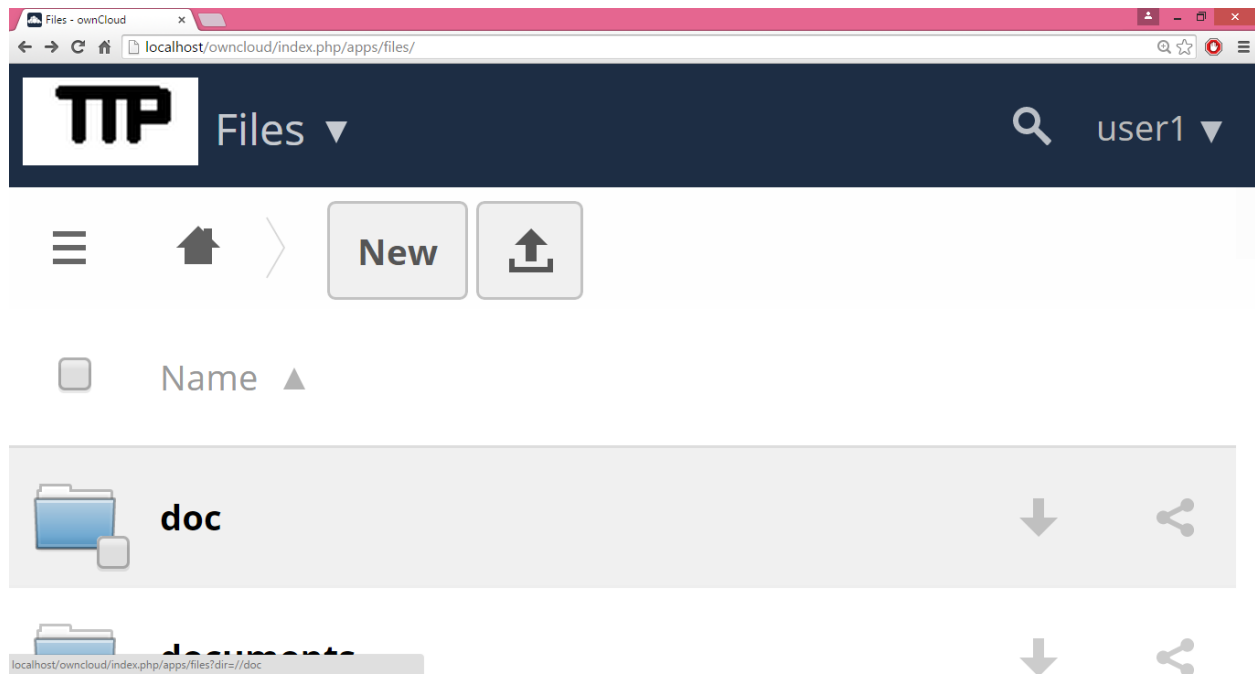
Adding text:



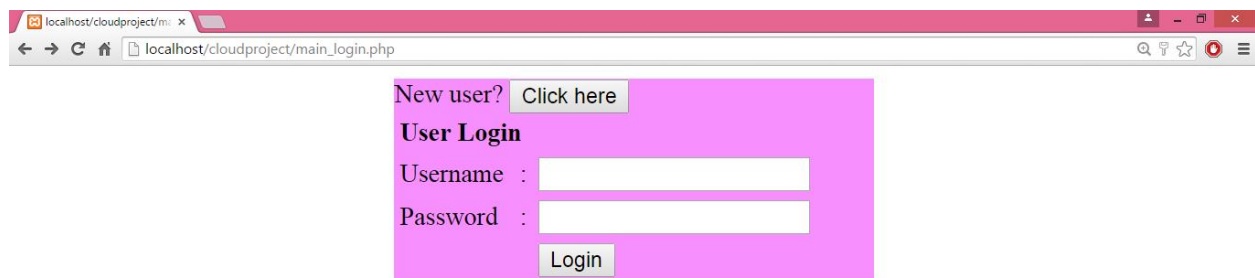
Edited file:



TTP option in dashboard:



TTP login:





Registering new user in TTP:

A screenshot of a web browser window with the address bar showing `localhost/cloudproject/signup.php?`. The page contains a registration form with the following fields and a button:

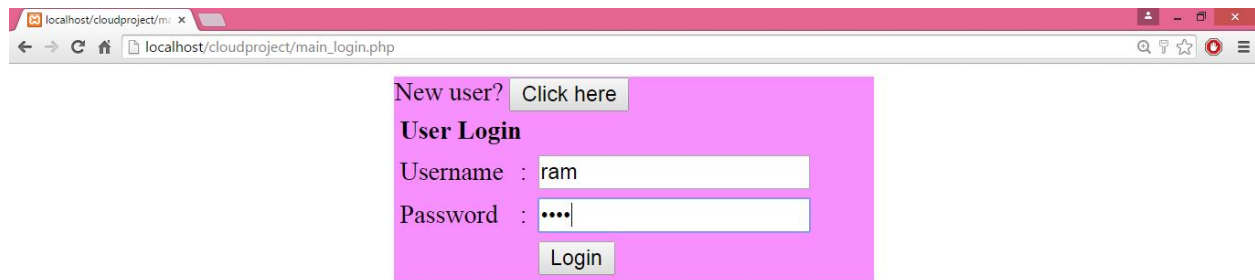
Name	<input type="text"/>
Email	<input type="text"/>
UserName	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Sign-Up"/>	

Signup for new user:

A screenshot of a web browser window with the address bar showing `localhost/cloudproject/signup.php?`. The page contains a registration form with the following fields and a button, where the first three fields are pre-filled:

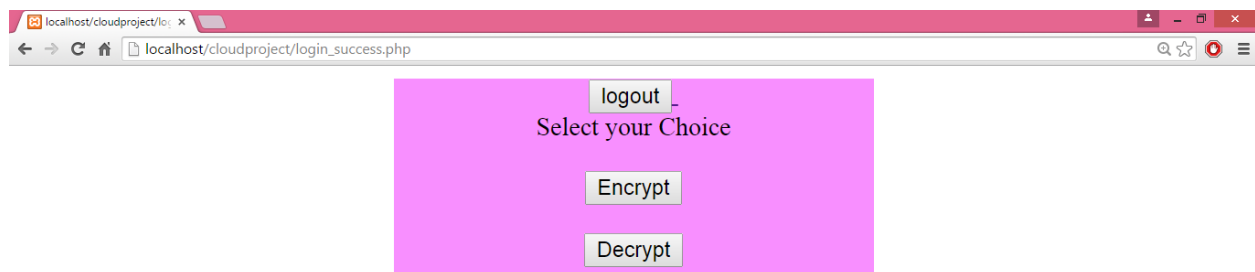
Name	<input type="text" value="ram"/>
Email	<input type="text" value="ramkumar@gmail.com"/>
UserName	<input type="text" value="ram"/>
Password	<input type="password" value="...."/>
<input type="button" value="Sign-Up"/>	

## User Login:



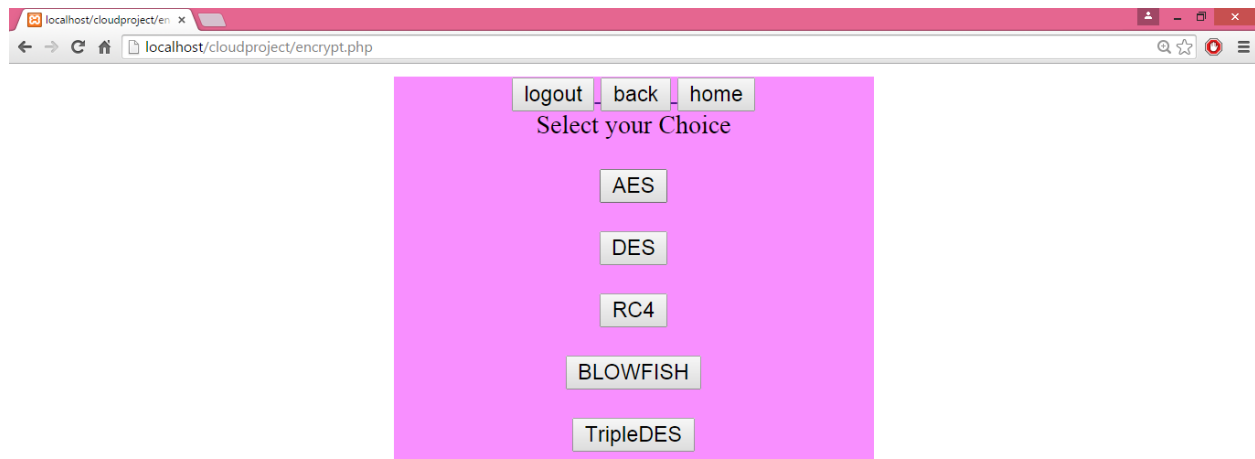
A screenshot of a web browser window. The address bar shows the URL `localhost/cloudproject/main_login.php`. The page content is on a light blue background. It features a link "New user? Click here" with a button. Below this is the "User Login" section, which includes a "Username :" label followed by a text input containing "ram", and a "Password :" label followed by a password input containing four dots. A "Login" button is positioned below the password field.

## Options in TTP:



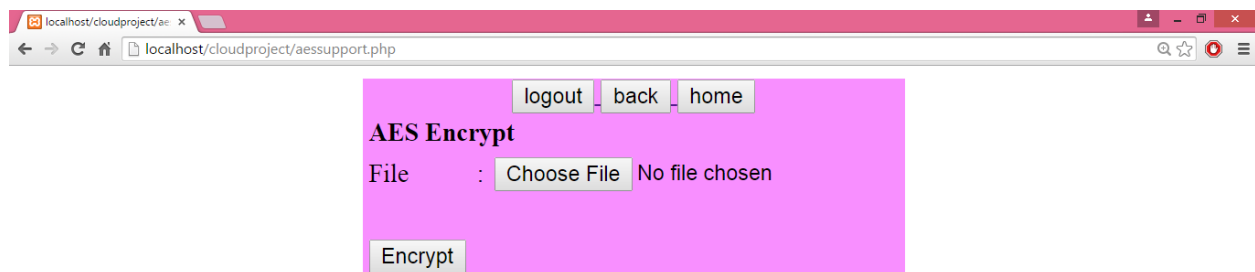
A screenshot of a web browser window. The address bar shows the URL `localhost/cloudproject/login_success.php`. The page content is on a light blue background. It features a "logout" button at the top. Below it is the text "Select your Choice". Underneath this text are two buttons: "Encrypt" and "Decrypt".

List for algorithm:

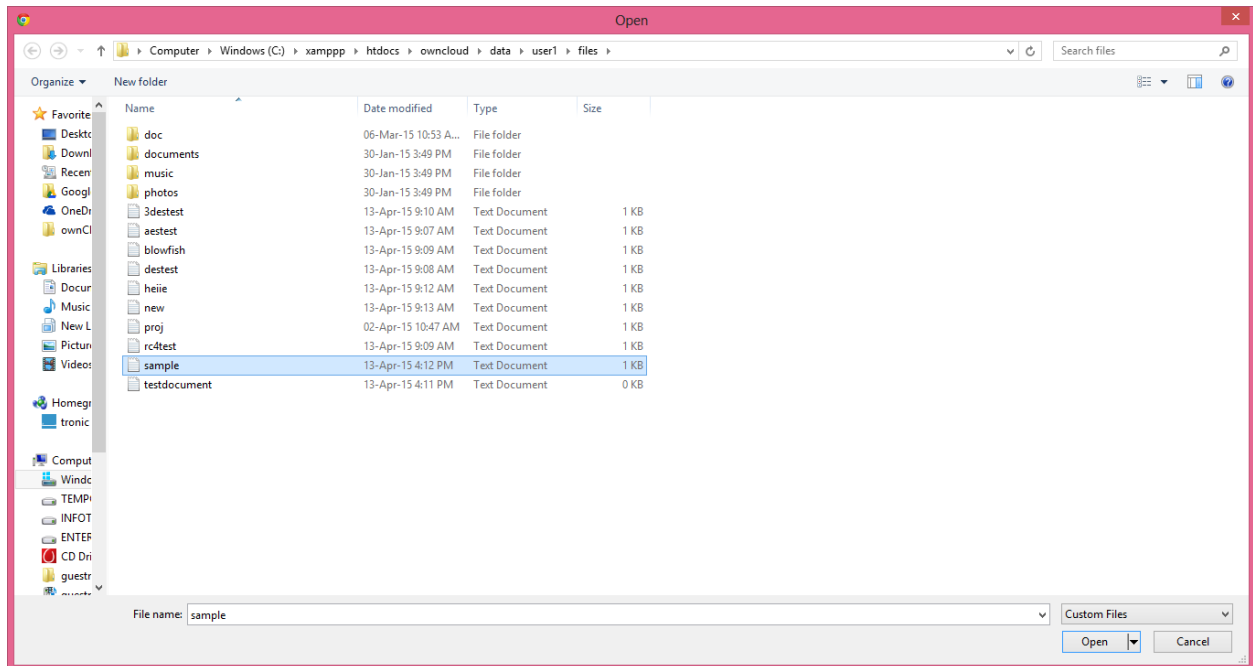


localhost/cloudproject/aessupport.php

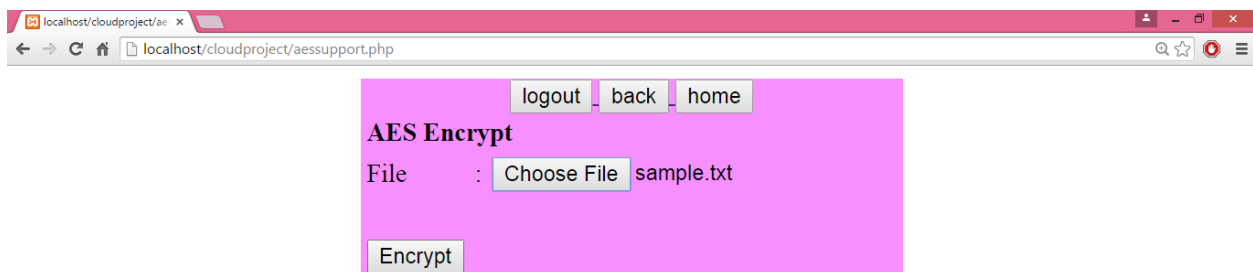
Encryption using AES:



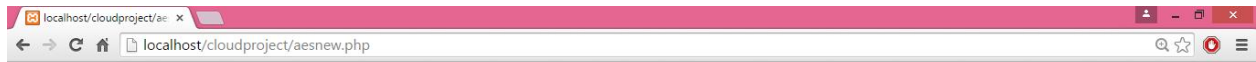
## File Upload:



## Uploaded file:

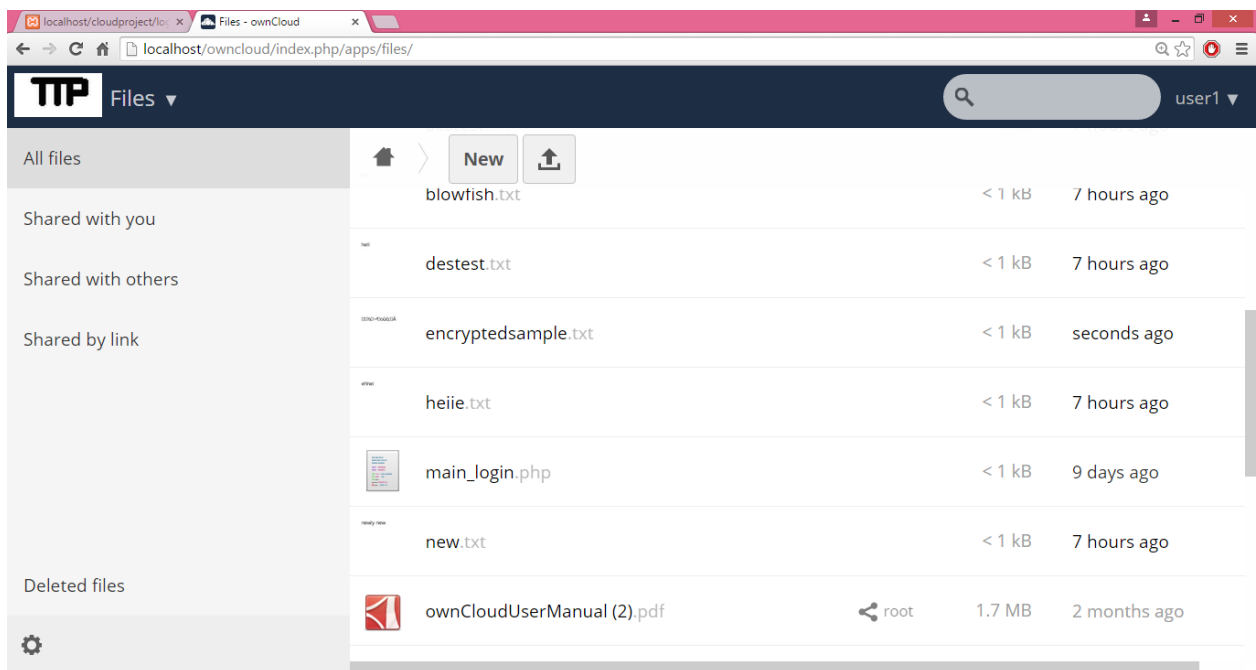


Encrypted file name:

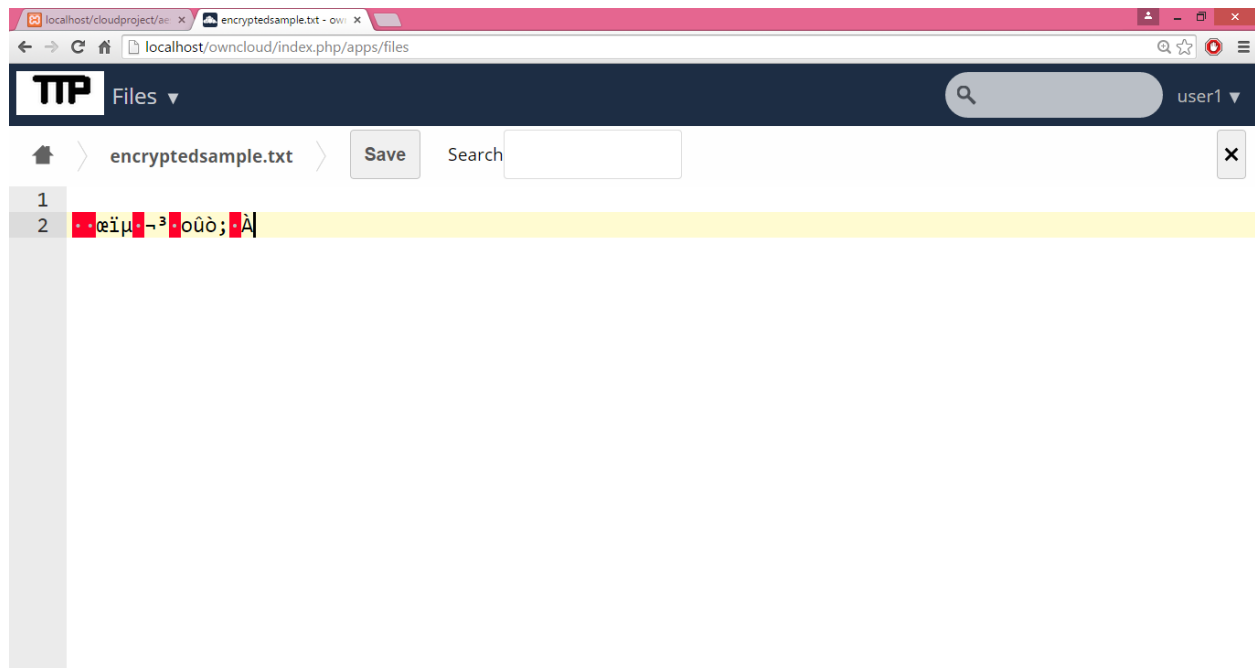


your file name is :encryptedsample.txt

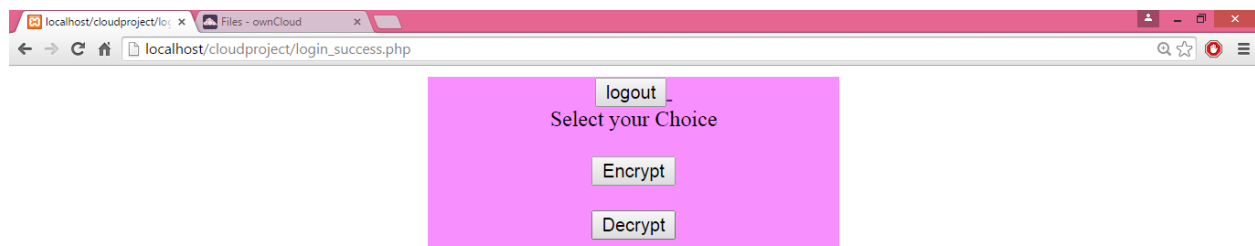
Encrypted file in dashboard:



Encrypted content:

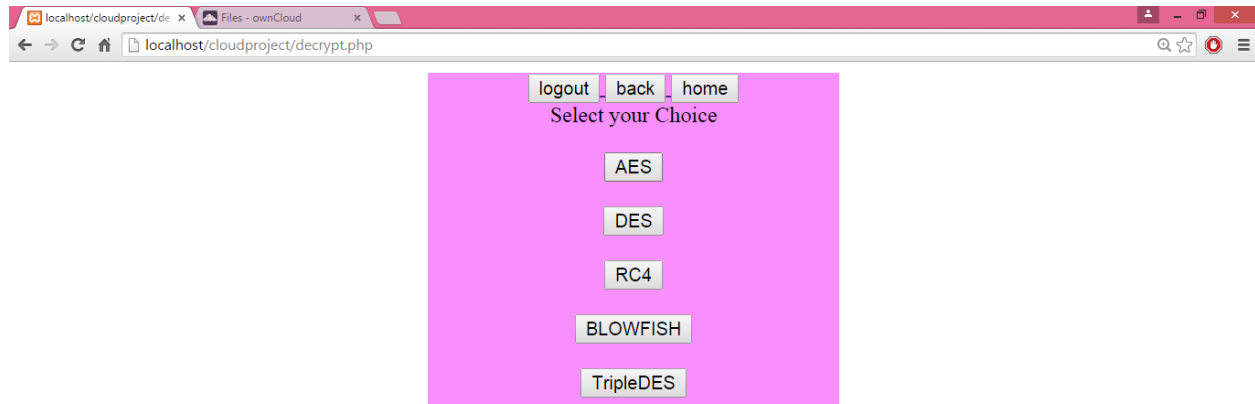


Options inTTP:



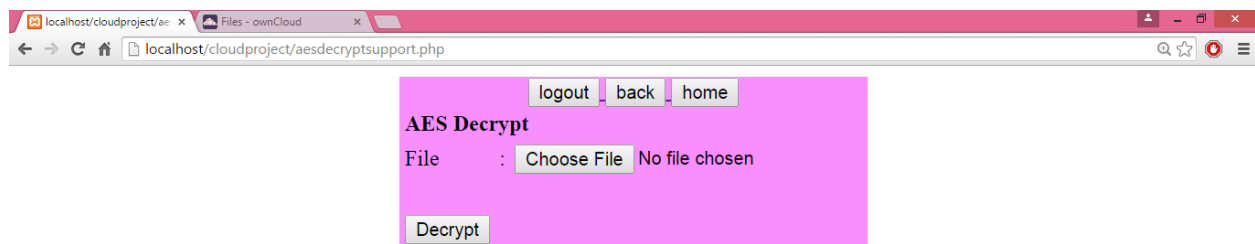
localhost/cloudproject/decrypt.php

## List of options for Decryption:

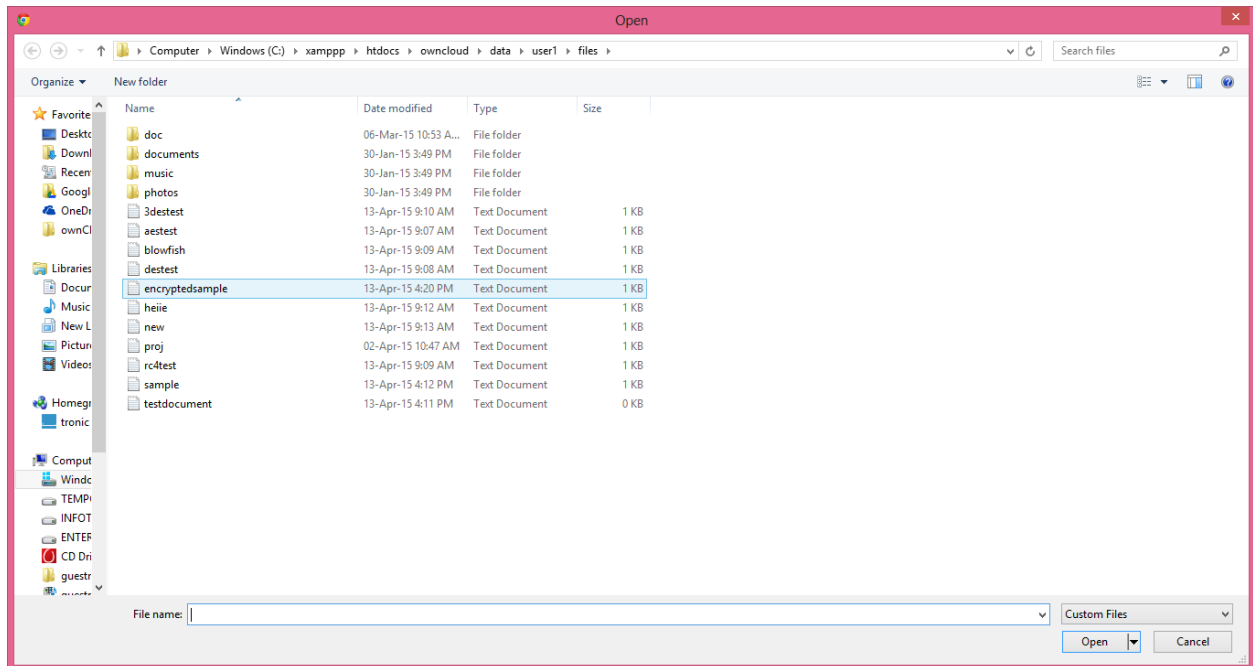


localhost/cloudproject/aesdecryptsupport.php

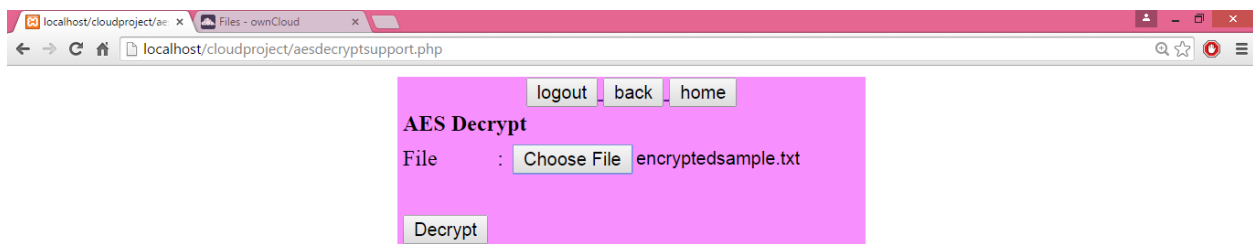
## AES decryption :



File upload for decryption:

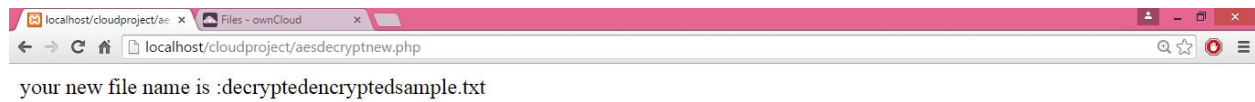


Uploaded file for decryption:

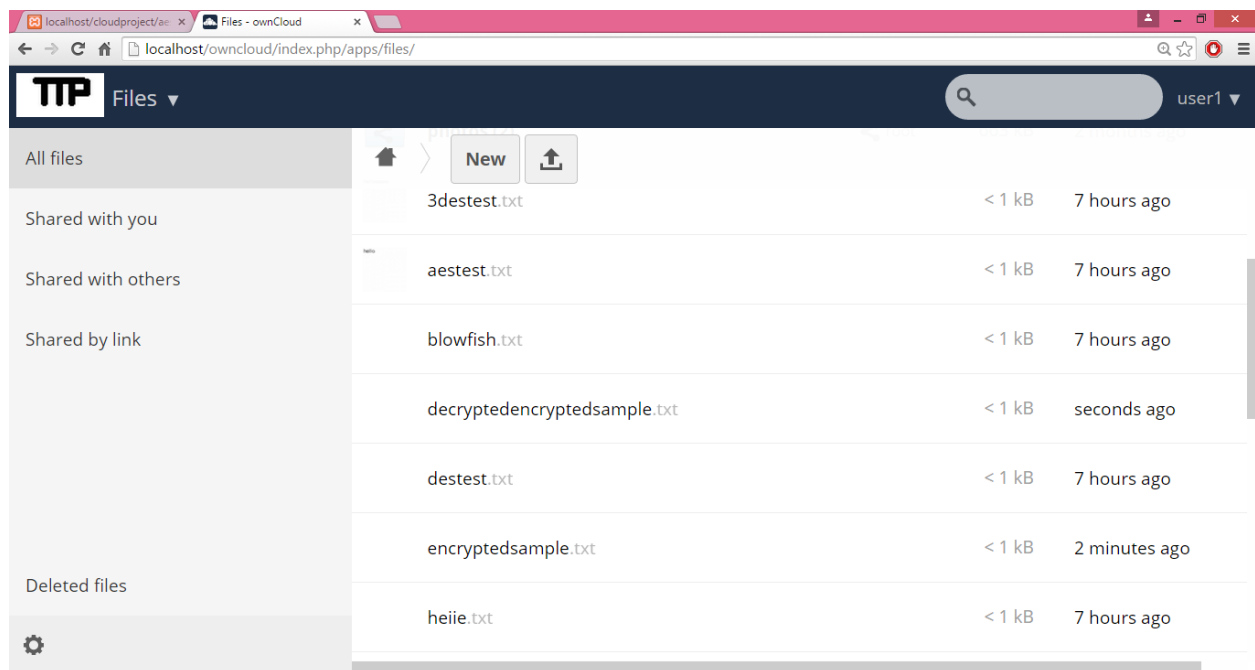




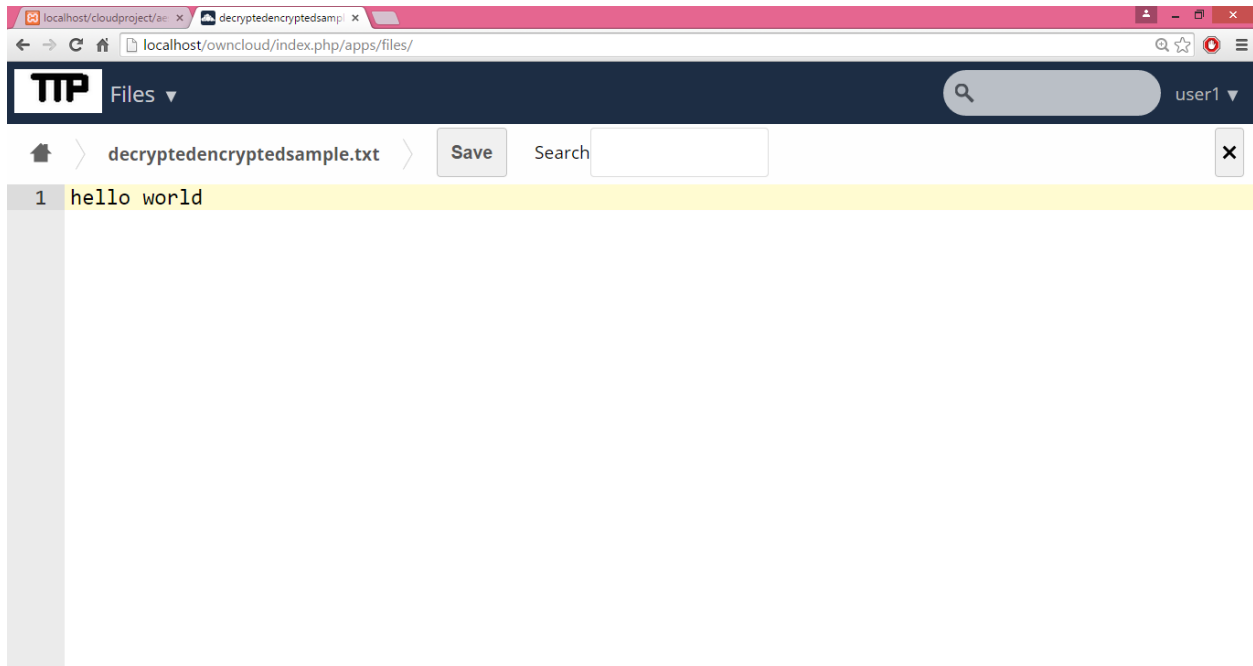
## Decrypted file:



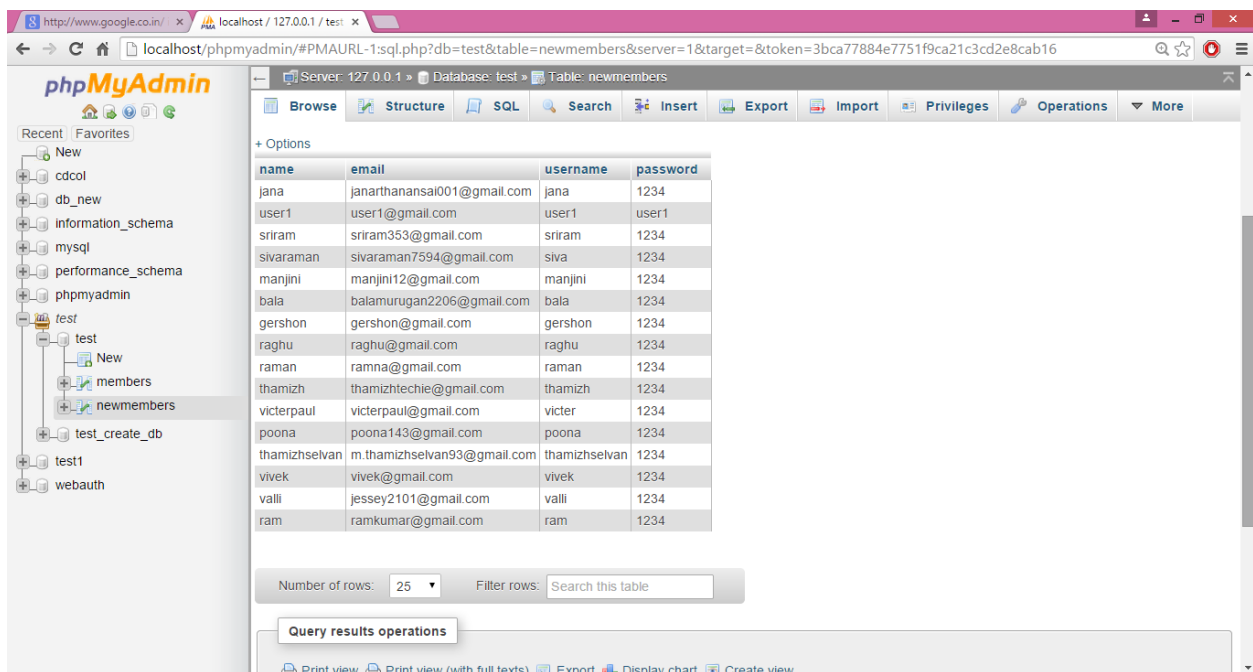
## Decrypted file in dashboard:



Decrypted content:



Database:



## APPENDIX II

### SAMPLE CODE

TripleDESSupport.php

```
<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#F88FFF">
<tr>

<form name="form1" method="post" action="TripleDES.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">
<tr>
<center>
<a href=main_login.php> <input type=button value=logout name=logout> </a>
<a href=encrypt.php> <input type=button value=back name=back> </a>
<a href=login_success.php> <input type=button value=home name=home> </a>
</center>
<td colspan="3"><strong>Triple DES Encrypt </strong></td>
</tr>
<tr>

<td width="78">File</td>
<td width="20">:</td>
<td width="294"><input name="filesource" type="file" accept="text/*" id="filesource"></td>

</tr>

<tr>
<td>&nbsp;</td>
```

```

<td>&nbsp;</td>
<tr> <td> <input type="submit" name= "Encrypt" value="Encrypt"></td></tr>
</tr>
</script>
</form>

```

### TripleDESdecryptsupport.php

```

<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#F88FFF">
<tr>
<form name="form1" method="post" action="TripleDESdecrypt.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">
<tr>
<center>
<a href=main_login.php> <input type=button value=logout name=logout> </a>
<a href=decrypt.php> <input type=button value=back name=back> </a>
<a href=login_success.php> <input type=button value=home name=home> </a>
</center>
<td colspan="3"><strong>Triple DES Decrypt </strong></td>
</tr>
<tr>
<td width="78">File</td>
<td width="20">:</td>
<td width="294"><input name="filesource" type="file" accept="text/*" id="filesource"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<tr> <td> <input type="submit" name= "Decrypt" value="Decrypt"></td></tr>

```

</tr>

</script>

</form>

TripleDESdecrypt.php

<?php

```
include 'Crypt/TripleDES.php';

$nameoffile=$_POST['filesource'];

$default= "decrypted";

$newfile= $default. "" . $nameoffile;

$content = file_get_contents("http://localhost/cloudproject/mirror/$nameoffile");

echo " your file name is :";

echo "$newfile";

$des = new Crypt_TripleDES();

$des->setKey('password');

$size = 10 * 1024;

$plaintext = "$content";

for ($i = 0; $i < $size; $i++)

{

    $plaintext.= "";

}

$stuff = $des->decrypt($plaintext);

file_put_contents("mirror/$newfile",$stuff);

?>
```

TripleDES.php

<?php

```
include 'Crypt/TripleDES.php';

$nameoffile=$_POST['filesource'];

$default= "encrypted";

$newfile= $default. "" . $nameoffile;

echo " your file name is :";

echo "$newfile";
```

```

$content = file_get_contents("http://localhost/cloudproject/mirror/$nameoffile");
$des = new Crypt_TripleDES();
$des->setKey('password');
$size = 10 * 1024;
$plaintext = "$content";
for ($i = 0; $i < $size; $i++)
{
    $plaintext.= " ";
}
$stuff = $des->encrypt($plaintext);
file_put_contents("mirror/$newfile",$stuff);
?>

```

#### signup.php

```

<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#C88CCC">
<tr>
<form name="form1" method="post" action="checksignup.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">
<td>Name</td><td> <input type="text" name="name" id="name"></td>
</tr>
<tr>
<td>Email</td><td> <input type="text" name="email" id="email"></td>
</tr>
<tr>
<td>UserName</td><td> <input type="text" name="user" id="user"></td>
</tr>
<tr>
<td>Password</td><td> <input type="password" name="pass" id="pass"></td>
</tr>
<tr>

```

```

<td><input id="button" type="submit" name="submit" value="Sign-Up"></td>
</tr>
</td>

```

rsasupport.php

```

<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#F88FFF">
<tr>
<form name="form1" method="post" action="rsa.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">
<tr>
<center>
<a href=main_login.php> <input type=button value=logout name=logout> </a>
<a href=encrypt.php> <input type=button value=back name=back> </a>
<a href=login_success.php> <input type=button value=home name=home> </a>
</center>
<td colspan="3"><strong>RSA Encrypt </strong></td>
</tr>
<tr>
<td width="78">File</td>
<td width="20">:</td>
<td width="294"><input name="filesource" type="file" accept="text/*" id="filesource"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>

```

```

<tr> <td> <input type="submit" name= "Encrypt" value="Encrypt"></td></tr>
</tr>
</script>
</form>

```

rsadecryptsupport.php

```

<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#F88FFF">

```

```

<tr>

```

```

<form name="form1" method="post" action="rsadecrypt.php">

```

```

<td>

```

```

<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">

```

```

<tr>

```

```

<center>

```

```

<a href=main_login.php> <input type=button value=logout name=logout> </a>

```

```

<a href=decrypt.php> <input type=button value=back name=back> </a>

```

```

<a href=login_success.php> <input type=button value=home name=home> </a>

```

```

</center>

```

```

<td colspan="3"><strong>RSA Decrypt </strong></td>

```

```

</tr>

```

```

<tr>

```

```

<td width="78">File</td>

```

```

<td width="20">:</td>

```

```

<td width="294"><input name="filesource" type="file" accept="text/*" id="filesource"></td>

```

```

</tr>

```



```

<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<tr> <td> <input type="submit" name= "Decrypt" value="Decrypt"></td></tr>

</tr>

</script>

</form>

```

rsadecrypt.php

```

<?php
    include 'Crypt/RSA.php';
    $nameoffile=$_POST['filesource'];
        $default= "decrypted";
        $newfile= $default. "" . $nameoffile;
        echo " your new file name is :";
        echo "$newfile";
        echo "<br>";
        $content = file_get_contents("$nameoffile");
    $rsa = new Crypt_RSA();
    extract($rsa->createKey());

    $plaintext = "$content";

    $rsa->loadKey($privatekey);
    $ciphertext = $rsa->encrypt($plaintext);

    $rsa->loadKey($publickey);

```

```

        echo "your decrypted content : ";
    echo $rsa->decrypt($ciphertext);
    $myfile = fopen("$newfile", "w") or die("Unable to open file!");
    fwrite($myfile, $rsa->decrypt($ciphertext));
?>

```

### rsa.php

```

<?php
include 'Crypt/RSA.php';
$nameoffile=$_POST['filesource'];
    $default= "encrypted";
    $newfile= $default. "" . $nameoffile;
    echo " your file name is :";
    echo "$newfile";
    echo "<br>";
    echo " your encrypted content : ";
    $content = file_get_contents("$nameoffile");
    $rsa = new Crypt_RSA();
    extract($rsa->createKey());

    $plaintext = "$content";

    $rsa->loadKey($privatekey);
    $ciphertext = $rsa->encrypt($plaintext);

    $rsa->loadKey($publickey);
    echo $ciphertext;
    $myfile = fopen("$newfile", "w") or die("Unable to open file!");
    fwrite($myfile, $ciphertext);
?>

```

rc4support.php

```
<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#F88FFF">
<tr>

<form name="form1" method="post" action="rc4.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">
<tr>
<center>
<a href=main_login.php> <input type=button value=logout name=logout> </a>
<a href=encrypt.php> <input type=button value=back name=back> </a>
<a href=login_success.php> <input type=button value=home name=home> </a>
</center>
<td colspan="3"><strong>RC4 Encrypt </strong></td>
</tr>
<tr>

<td width="78">File</td>
<td width="20">:</td>
<td width="294"><input name="filesource" type="file" accept="text/*" id="filesource"></td>

</tr>

<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<tr> <td> <input type="submit" name= "Encrypt" value="Encrypt"></td></tr>
```

</tr>

</script>

</form>

rc4decryptsupport.php

<table width="300" border="0" align="center" cellpadding="0" cellspacing="1" bgcolor="#F88FFF">

<tr>

<form name="form1" method="post" action="rc4decrypt.php">

<td>

<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">

<tr>

<center>

<a href=main\_login.php> <input type=button value=logout name=logout> </a>

<a href=decrypt.php> <input type=button value=back name=back> </a>

<a href=login\_success.php> <input type=button value=home name=home> </a>

</center>

<td colspan="3"><strong>RC4 Decrypt </strong></td>

</tr>

<tr>

<td width="78">File</td>

<td width="20">:</td>

<td width="294"><input name="filesource" type="file" accept="text/\*" id="filesource"></td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

<tr> <td> <input type="submit" name= "Decrypt" value="Decrypt"></td></tr>

</tr>

</script>

</form>

rc4decrypt.php

<?php

include 'Crypt/RC4.php';

\$nameoffile=\$\_POST['filesource'];

\$default= "decrypted";

\$newfile= \$default. "" . \$nameoffile;

\$content = file\_get\_contents("http://localhost/cloudproject/mirror/\$nameoffile");

echo " your file name is :";

echo "\$newfile";

\$rc4 = new Crypt\_RC4();

\$rc4->setKey('password');

\$size = 10 \* 1024;

\$plaintext = "\$content";

for (\$i = 0; \$i < \$size; \$i++)

{

\$plaintext.= ";

}

\$stuff = \$rc4->decrypt(\$plaintext);

```
file_put_contents("mirror/$newfile",$stuff);
```

```
?>
```

#### Rc4.php

```
<?php
```

```
include 'Crypt/RC4.php';
```

```
$nameoffile=$_POST['filesource'];
```

```
$default= "decrypted";
```

```
$newfile= $default. "" . $nameoffile;
```

```
$content = file_get_contents("http://localhost/cloudproject/mirror/$nameoffile");
```

```
echo " your file name is :";
```

```
echo "$newfile";
```

```
$rc4 = new Crypt_RC4();
```

```
$rc4->setKey('password');
```

```
$size = 10 * 1024;
```

```
$plaintext = "$content";
```

```
for ($i = 0; $i < $size; $i++)
```

```
{
```

```
    $plaintext.= ";
```

```
}
```

```
$stuff = $rc4->decrypt($plaintext);
```

```
file_put_contents("mirror/$newfile",$stuff);
```

```
?>
```

#### Mainlogin.php

```
<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#F88FFF">
```

```
<tr>
```

```
<form name="form1" method="post" action="checklogin.php">
```

```
<td>
```

```
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">
```

```
<tr>
```

```

<td colspan="3"><strong>User Login </strong></td>
</tr>
<tr>
<td width="78">Username</td>
<td width="6">:</td>
<td width="294"><input name="myusername" type="text" id="myusername"></td>
</tr>
<tr>
<td>Password</td>
<td>:</td>
<td><input name="mypassword" type="password" id="mypassword"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td><input type="submit" name="Submit" value="Login"></td>
</tr>
</form>
<form method="get" action="signup.php">
  New user? <button type="submit">Click here</button>
</form>
</tr>
</table>
</td>
</form>
</tr>
</table>

```

## Login\_success.php

```
<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#F88FFF" >
<tr>
<form name="form11">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">
<tr>
<td colspan="3"><strong>
</td>
</tr>
<center>
<a href=main_login.php> <input type=button value=logout name=logout > </a> <br>
</center>
<tr><center>Select your Choice</center><br>
<a href=encrypt.php> <center> <input type=button value=Encrypt name=encrypt > </center>
<br>
<a href=decrypt.php> <center> <input type=button value=Decrypt name=decrypt > </center>
</tr>
</tr>

</form>
</form>
</form>
```



encryptidentifier.php

```
<?php
$algo= $_POST['algo'];
$dataa= $_POST['filesource'];
$content = file_get_contents("");
if($algo=='AES')
{
header("location:aes.php");
}
if($algo=='DES')
{
header("location:des.php");
}
if($algo=='RSA')
{
header("location:rsa.php");
}if($algo=='BLOWFISH')
{
header("location:blowfish.php");
}
?>
```

## Encrypt.php

```
<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#F88FFF" >
<tr>
<form name="form11">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">
<tr>
<td colspan="3"><strong>
</td>
</tr>
<center>
<a href=main_login.php> <input type=button value=logout name=logout> </a>
<a href=login_success.php> <input type=button value=back name=back> </a>
<a href=login_success.php> <input type=button value=home name=home> </a>
</center>
<tr><center>Select your Choice</center><br>
<a href=aessupport.php> <center> <input type=button value=AES name=AES > </center> <br>
<a href=dessupport.php> <center> <input type=button value=DES name=DES > </center> <br>
<a href=rc4support.php> <center> <input type=button value=RC4 name=RC4 > </center> <br>
<a href=blowfishsupport.php> <center> <input type=button value=BLOWFISH
name=BLOWFISH > </center> <br>
<a href=TripleDESSupport.php> <center> <input type=button value=TripleDES
name=TripleDES > </center>

</tr>
</tr>
</form>
```

## Dessupport.php

```
<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#F88FFF">
<tr>
<form name="form1" method="post" action="des.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">
<tr>
<center>
<a href=main_login.php> <input type=button value=logout name=logout> </a>
<a href=encrypt.php> <input type=button value=back name=back> </a>
<a href=login_success.php> <input type=button value=home name=home> </a>
</center>
<td colspan="3"><strong>DES Encrypt </strong></td>
</tr>
<tr>
<td width="78">File</td>
<td width="20">:</td>
<td width="294"><input name="filesource" type="file" accept="text/*" id="filesource"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<tr> <td> <input type="submit" name= "Encrypt" value="Encrypt"></td></tr>
</tr>
</script>
</form>
```

desdecryptsupport.php

```
<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#F88FFF">
<tr>
<form name="form1" method="post" action="desdecrypt.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">
<tr>
<center>
<a href=main_login.php> <input type=button value=logout name=logout> </a>
<a href=decrypt.php> <input type=button value=back name=back> </a>
<a href=login_success.php> <input type=button value=home name=home> </a>
</center>
<td colspan="3"><strong>DES Decrypt </strong></td>
</tr>
<tr>
<td width="78">File</td>
<td width="20">:</td>
<td width="294"><input name="filesource" type="file" accept="text/*" id="filesource"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<tr> <td> <input type="submit" name= "Decrypt" value="Decrypt"></td></tr>
</tr>
</script>
</form>
```

## desdecrypt.php

```
<?php
    include 'Crypt/DES.php';
    $nameoffile=$_POST['filesource'];
    $default= "decrypted";
    $newfile= $default. "" . $nameoffile;
    $content = file_get_contents("http://localhost/cloudproject/mirror/$nameoffile");
    echo " your new file name is :";
    echo "$newfile";
    $des = new Crypt_DES();
    $des->setKey('password');
    $size = 10 * 1024;
    $plaintext = "$content";
    for ($i = 0; $i < $size; $i++)
    {
        $plaintext.= "";
    }
    $stuff = $des->decrypt($plaintext);
    file_put_contents("mirror/$newfile",$stuff);
?>
```

## Des.php

<?php

```
include 'Crypt/DES.php';

$nameoffile=$_POST['filesource'];
$default= "encrypted";
$newfile= $default. "" . $nameoffile;
echo " your file name is :";
echo "$newfile";

$content = file_get_contents("http://localhost/cloudproject/mirror/$nameoffile");
$des = new Crypt_DES();
$des->setKey('password');
$size = 10 * 1024;
$plaintext = "$content";
for ($i = 0; $i < $size; $i++)
{
    $plaintext.= "";
}
$stuff = $des->encrypt($plaintext);
file_put_contents("mirror/$newfile",$stuff);
```

?>

decryptidentifier.php

```
<?php
$algo= $_POST['algo'];

if($algo=='AES')
{
header("location:aesdecrypt.php");
}
if($algo=='DES')
{
header("location:desdecrypt.php");
}
if($algo=='RSA')
{
header("location:rsadecrypt.php");
}if($algo=='BLOWFISH')
{
header("location:blowfishdecrypt.php");
}
?>
```

## Decrypt.php

```
<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#F88FFF" >
<tr>
<form name="form11">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">
<tr>
<td colspan="3"><strong>
</td>
</tr>
<center>
<a href=main_login.php> <input type=button value=logout name=logout> </a>
<a href=login_success.php> <input type=button value=back name=back> </a>
<a href=login_success.php> <input type=button value=home name=home> </a>
</center>
<tr><center>Select your Choice</center><br>
<a href=aesdecryptsupport.php> <center> <input type=button value=AES name=AES >
</center> <br>
<a href=desdecryptsupport.php> <center> <input type=button value=DES name=DES >
</center> <br>
<a href=rc4decryptsupport.php> <center> <input type=button value=RC4 name=RC4 >
</center> <br>
<a href=blowfishdecryptsupport.php> <center> <input type=button value=BLOWFISH
name=BLOWFISH > </center> <br>
<a href=TripleDESdecryptsupport.php> <center> <input type=button value=TripleDES
name=TripleDES >

</tr>
</tr>
</form>
```



### ChecksSignup.php

```
<?php
$host="localhost"; // Host name
$username=""; // Mysql username
$password=""; // Mysql password
$db_name="test"; // Database name
$tbl_name="newmembers"; // Table name
// Connect to server and select database.
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");
$name= $_POST['name'];
$email= $_POST['email'];
$user= $_POST['user'];
$pass= $_POST['pass'];
$check = mysql_query("select * from newmembers where username='$user'") or
die(mysql_error());
$count = mysql_num_rows($check);
if ($count > 0)
{
    $msg='Username Already Exist!';
}
else
{
    $sql = mysql_query("INSERT INTO newmembers (name,email,username,password)
VALUES ('$name','$email','$user','$pass')");

}
header("location:main_login.php");

?>
```

checklogin.php

<?php

```
$host="localhost"; // Host name
$username=""; // Mysql username
$password=""; // Mysql password
$db_name="test"; // Database name
$tbl_name="newmembers"; // Table name
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");
// username and password sent from form
$myusername=$_POST['myusername'];
$mypassword=$_POST['mypassword'];
$sql="SELECT * FROM $tbl_name WHERE username='$myusername' and
password='$mypassword'";
$result=mysql_query($sql);
$count=mysql_num_rows($result);

if($count==1){

// Register $myusername, $mypassword and redirect to file "login_success.php"

header("location:login_success.php");
}
else {
echo "Wrong Username or Password";
}
?>
```

blowfishsupport.php

```
<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#F88FFF">
<tr>
<form name="form1" method="post" action="blowfish.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">
<tr>
<center>
<a href=main_login.php> <input type=button value=logout name=logout> </a>
<a href=encrypt.php> <input type=button value=back name=back> </a>
<a href=login_success.php> <input type=button value=home name=home> </a>
</center>
<td colspan="3"><strong>BLOWFISH Encrypt </strong></td>
</tr>
<tr>
<td width="78">File</td>
<td width="20">:</td>
<td width="294"><input name="filesource" type="file" accept="text/*" id="filesource"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<tr> <td> <input type="submit" name= "Encrypt" value="Encrypt"></td></tr>
</tr>
</script>
</form>
```

### Blowfishdecryptsupport.php

```
<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#F88FFF">
<tr>
<form name="form1" method="post" action="blowfishdecrypt.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">
<tr>
<center>
<a href=main_login.php> <input type=button value=logout name=logout> </a>
<a href=decrypt.php> <input type=button value=back name=back> </a>
<a href=login_success.php> <input type=button value=home name=home> </a>
</center>
<td colspan="3"><strong>BLOWFISH Decrypt </strong></td>
</tr>
<tr>
<td width="78">File</td>
<td width="20">:</td>
<td width="294"><input name="filesource" type="file" accept="text/*" id="filesource"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<tr> <td> <input type="submit" name= "Decrypt" value="Decrypt"></td></tr>
</tr>
</script>
</form>
```

### Blowfishdecrypt.php

```
<?php
include 'Crypt/Blowfish.php';
$nameoffile=$_POST['filesource'];
    $default= "decrypted";
    $newfile= $default. "" . $nameoffile;
    $content = file_get_contents("http://localhost/cloudproject/mirror/$nameoffile");
    echo " your new file name is :";
    echo "$newfile";

$blowfish = new Crypt_Blowfish();
$blowfish->setKey('12345678901234567890123456789012');
$plaintext = "$content";
    $stuff = $blowfish->decrypt($plaintext);
    file_put_contents("mirror/$newfile",$stuff);

?>
```

### Blowfish.php

```
<?php
include 'Crypt/Blowfish.php';
$nameoffile=$_POST['filesource'];
    $default= "encrypted";
    $newfile= $default. "" . $nameoffile;
    echo " your file name is :";
    echo "$newfile";

    $content = file_get_contents("http://localhost/cloudproject/mirror/$nameoffile");
$blowfish = new Crypt_Blowfish();
$blowfish->setKey('12345678901234567890123456789012');
$plaintext = "$content";
    $stuff = $blowfish->encrypt($plaintext);
    file_put_contents("mirror/$newfile",$stuff);

?>
```

## Aessupport.php

```
<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#F88FFF">
<tr>
<form name="form1" method="post" action="aesnew.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">
<tr>
<center>
<a href=main_login.php> <input type=button value=logout name=logout> </a>
<a href=encrypt.php> <input type=button value=back name=back> </a>
<a href=login_success.php> <input type=button value=home name=home> </a>
</center>
<td colspan="3"><strong>AES Encrypt </strong></td>
</tr>
<tr>
<td width="78">File</td>
<td width="20">:</td>
<td width="294"><input name="filesource" type="file" accept="text/*" id="filesource"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<tr> <td> <input type="submit" name= "Encrypt" value="Encrypt"></td></tr>
</tr>
</script>
</form>
```

### Aesnew.php

```
<?php
include 'Crypt/AES.php';
$nameoffile=$_POST['filesource'];
$default= "encrypted";
$newfile= $default. "" . $nameoffile;
echo " your file name is :";
echo "$newfile";
$content = file_get_contents("http://localhost/cloudproject/mirror/$nameoffile");
$aes = new Crypt_AES();
$aes->setKey('password');
$size = 10 * 1024;
$plaintext = "$content";
for ($i = 0; $i < $size; $i++)
{
    $plaintext.= "";
}
$stuff = $aes->encrypt($plaintext);
file_put_contents("mirror/$newfile",$stuff);
?>
```

Aesdecryptsupport.php

```
<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#F88FFF">
<tr>
<form name="form1" method="post" action="aesdecryptnew.php">
<td>
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#F88FFF">
<tr>
<center>
<a href=main_login.php> <input type=button value=logout name=logout> </a>
<a href=decrypt.php> <input type=button value=back name=back> </a>
<a href=login_success.php> <input type=button value=home name=home> </a>
</center>
<td colspan="3"><strong>AES Decrypt </strong></td>
</tr>
<tr>
<td width="78">File</td>
<td width="20">:</td>
<td width="294"><input name="filesource" type="file" accept="text/*" id="filesource"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<tr> <td> <input type="submit" name= "Decrypt" value="Decrypt"></td></tr>
</tr>
</script>
</form>
```



## Aesdecryptnew.php

<?php

```
include 'Crypt/AES.php';

$nameoffile=$_POST['filesource'];
$default= "decrypted";
$newfile= $default. "" . $nameoffile;
$content = file_get_contents("http://localhost/cloudproject/mirror/$nameoffile");
echo " your new file name is :";
echo "$newfile";

$aes = new Crypt_AES();
$aes->setKey('password');

$size = 10 * 1024;
$plaintext = "$content";
for ($i = 0; $i < $size; $i++)
{
    $plaintext.= "";
}

$stuff = $aes->decrypt($plaintext);
file_put_contents("mirror/$newfile",$stuff);
```

?>

## **APPENDIX III**

### **PUBLICATIONS**

- M.Thamizhselvan, S.Gershon manoj, R.Raghuraman, P.Victer Paul," A NOVEL SECURITY MODEL FOR CLOUD USING TRUSTED THIRD PARTY ENCRYPTION", IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems ICIIECS'15, ISBN 978-1-4799-6818-3 (Scopus indexed)
- M.Thamizhselvan, S.Gershon manoj, R.Raghuraman, P.Victer Paul,"DATA SECURITY MODEL FOR CLOUD COMPUTING USING V-GRT METHODOLOGY", IEEE Sponsored 9th International Conference on Intelligent Systems and Control (ISCO)2015, ISBN 978-1-4799-6480-2(Scopus indexed)
- M.Thamizhselvan, S.Gershon manoj, R.Raghuraman, P.Victer Paul,"DATA SECURITY MODEL FOR CLOUD COMPUTING USING V-GRT METHODOLOGY", International Journal of Applied Engineering Research (IJAER), ISSN 0973-4562. (Paper selected)(Scopus indexed).