IMPROVE DATA SECURITY ON CLOUD COMPUTING USING HYBRID ALGORITHM

MISRAK FISEHA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINNERING SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING (SOEEC)

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

JULY, 2024

ADAMA, ETHIOPIA

IMPROVE DATA SECURITY ON CLOUD COMPUTINGUSING HYBRID ALGORITHM

MISRAK FISEHA TEKLEMARIAM

Under the guidance of DR. BIRUK YIRGA

Thesis

Submitted in partial fulfillment of the requirement for the degree of Master of Computer science and Engineering with focus area of Networking science and Security

Department of Computer Science and Engineering School of Electrical Engineering and Computing (SoEEC) Adama Science and Technology University (ASTU)

DECLARATION

I hereby declare that the thesis entitled **"Improve Data Security on Cloud Computing Using Hybrid Algorithm"** is submitted for Master Degree is my original work and the thesis work has not formed the basis for the award of any degree, associate ship, fellowship or any other similar titles.

Name of student	Signature	Place

CERTIFICATE

I, the major advisor/supervisor of this thesis, hereby certify that I have closely advised/supervised the student while developing this thesis and read the draft thesis entitled "Improve Data Security on Cloud Computing Using Hybrid Algorithm. Under my guidance by Dr. Biruk Yirga. Therefore, I recommend the submission of the proposal to the department for further review and evaluation

_____	_____	_____
Name of Advisor	Signature	Date

Approval by Members of Board of Reviewers

We, the undersigned, members of the Board of Reviewers of the thesis open defense by Misrak Fiseha (have read and evaluated the thesis entitled **"Improve Data Security on Cloud Computing Using Hybrid Algorithm"** and assessed the understanding of the candidate about the proposed research. This is, therefore, to certify that the thesis is accepted and we recommend the implementation of the thesis.

Chairperson	Signature	Date
Reviewer 1	Signature	Date
Reviewer 2	Signature	Date

Final approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head	Signature	Date
School Dean	Signature	Date
Office of Postgraduate Stu	Dean Signature	Date

List of Acronyms and Abbreviations

3DES	Triple Data Encryption Standard
AES	Advanced Encryption Standard
API	Application Program Interface
AWS	Amazon Web Services
CC	Cloud Computing
CPU	Central Processing Unit
CSP	Cloud Service Provider
DBMS	Database Management System
DES	Data Encryption Standard
DES	Data Encryption Standard
DF	Decrypted File
DFD	Data Flow Diagram
DK	Decrypted Key
EC2	Elastic Cloud Computing
EF	Encrypted File
E-mail	Electronic Mail
FH	File Hash
GFS	Google File System
GUI	Graphical User Interface
GUI	Graphical User Interface
HTML	Hyper Text Markup language
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
IaaS	Infrastructure as a Service

IBM	International Business Machine
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IMP	IP Multimedia Protocol
IT	Information Technology
Java API	The Java Application Programming Interface
Java EE	Java Platform Enterprise Edition
JDK	Java Development Kit
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
OS	Operating System
PaaS	Platform as a Service
PIL	Python Image Library
PSF	Python Software Foundation
QoS	Quality of Service
RDBMS	Relational Database Management System
RSA	Ravist Shamir Adelman
SaaS	Software as a Service
SMD	Smart Mobile Device
SQL	Structured Query Language
SSH	Secure Shell
TPA	Third Party Auditor
TPA	Third Party Auditor
UML	Unified Modeling Language
VMs	Virtual Machines
VPN	Virtual Private Network

Abstract

One of the significant advancements in information technology is Cloud computing, but the security issue data storage on file encryption and decryption AES key management is a big problem in the cloud environment. That is why a system is proposed in this thesis for improving the security of cloud data storage using encryption and hashing functions techniques deployed on heroku. Heroku to deploy, manage, and scale modern apps. Our platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market. In the data encryption phase, we implemented hybrid encryption using the algorithm of AES symmetric encryption and the algorithm of RSA asymmetric encryption and in the data validation phase, we use the SHA hashing algorithm. By using these methods it can achieve strong data security. These results increase the security protection techniques for end user and data owner from fake user and fake data owner in the cloud. To evaluate the performance of proposed schemes, we computed the security strength, time efficiency which is measured based on the upload time, key generation time, encryption time, file size after encryption and decryption time for different data files by the help of Python Prometheus. Prometheus uses time-series data with a timestamp and optional key-value pairs to store and retrieve metrics data.

Keywords: Data security, AES, RSA, file Hash, hybrid Algorithm.

Table of Contents

Contents

List of Figures

List of Tables

CHAPTER ONE

INTRODUCTION

1.1. Background of Study

Before Computing was come into existence, client Server Architecture was used where all the data and control of client resides in Server side. If a single user wants to access some data, firstly user need to connect to the server and after that user will get appropriate access. But it has many disadvantages. So, After Client Server computing, Distributed Computing was come into existence, in this type of computing all computers are networked together with the help of this, user can share their resources when needed. It also has certain limitations. So, in order to remove limitations faced in distributed system, cloud computing was emerged.

Cloud computing has recently surfaced as a new paradigm for hosting and delivering services over the web. Cloud computing has been envisioned as the next generation paradigm in computation. In the cloud computing environment, both applications and resources are delivered on demand over the Internet as services. Cloud is an environment of the hardware and software resources in the data centers that provide diverse services over the network or the Internet to satisfy user's requirements (N. Leavitt, 2009).

National Institute of Standards and Technology (NIST) is that cloud computing enables ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. According to the explanation, cloud computing provides a convenient on-demand network access to a shared pool of configurable computing resources. Resources refer to computing applications, network resources, platforms, software services, virtual servers, and computing infrastructure.

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

Infrastructure as a service (IaaS): This model provides on-demand access to cloud infrastructure, such as servers, storage, and networking. This eliminates the need to procure,

manage, and maintain on-premises infrastructure.

Platform as a service (PaaS): PaaS denotes cloud platforms that provide runtime environments for developing, testing and managing applications, examples of PaaS would include Heroku and Google App Engine

- **Software as a service (SaaS):** This model offers cloud-based applications that are delivered and maintained by the service provider, eliminating the need for end users to deploy software locally.

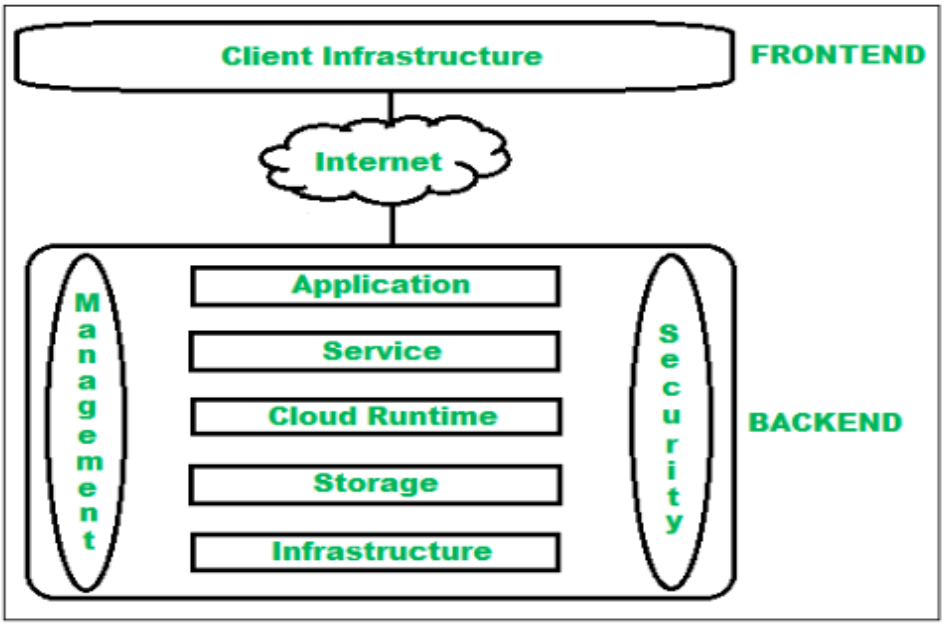


Figure 1. 1: Architecture of Cloud Computing (www.google.com 2019)

Need of Security in Cloud Security in cloud computing is crucial to any company looking to keep its applications and data protected from bad actors. Maintaining a strong cloud security posture helps organizations achieve the now widely recognized benefits of cloud computing. Cloud security comes with its own advantages as well, helping you achieve lower upfront costs, reduced ongoing operational and administrative costs, easier scaling, increased reliability and availability, and improved DDoS protection

Figure 1. 2: Security is a major concern to cloud computing.

(www.digitalguardian.com, 2017)

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

- On-demand self-services
- Resource pooling
- Measured service
- Virtualization
- Flexible pricing models

Data security Data security becomes particularly serious in the cloud computing environment, because data are distributed in different machines and storage devices including servers, PCs, and various mobile devices such as wireless sensor networks and smart phones. Data security in the cloud computing is more complicated than data security in the traditional information systems. If security measures are not achieved properly for data operation and transmission,

the data is at high risk. Strongest security measures are to be implemented by identifying security challenge and solution to handle these challenges.

Figures 1 show the data security challenges, and it is clear that Data leak prevention and data segregation and protection has more impact on security challenges.

Therefore, in order to ensure secure information and that the data are not attacked by security vulnerabilities, cloud computing should implementing additional safety measures, along with conventional safety controls to improve the security in cloud computing it is significant to offer authorization, authentication, and access control to data stored in cloud.

- Confidentiality: to ensure that data is protected from any attacks.
- Integrity: to ensure not store their private data like, password therefore that integrity can assured.
- Availability: it depends on the agreement between the vendor and the client.

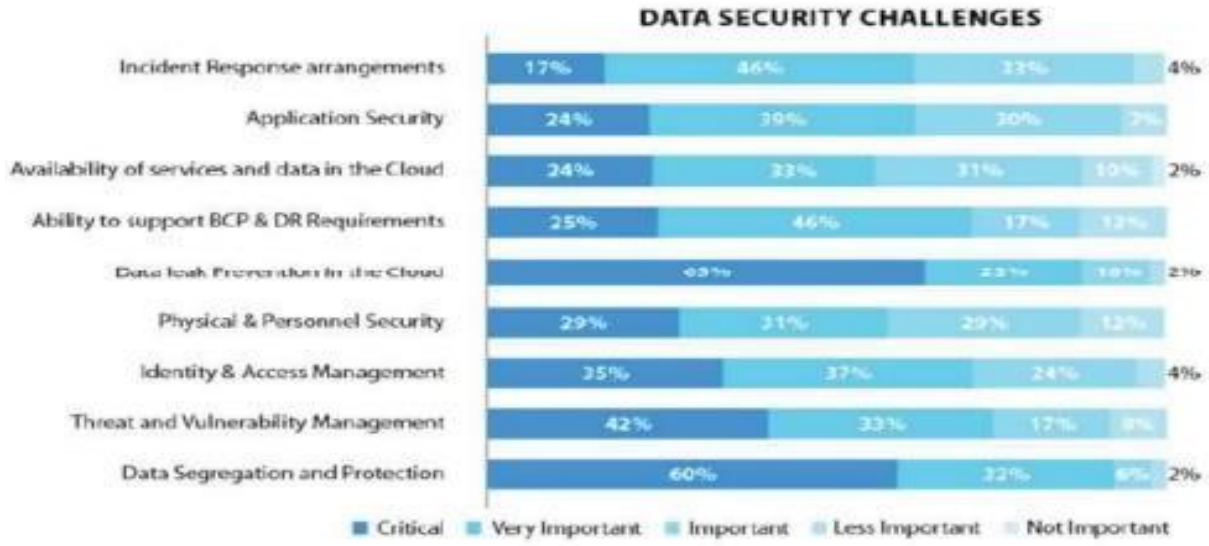


Figure 1. 3: Data security challenges

There are three types of cryptography

Symmetric key cryptography is also known as secret-key cryptography, and in this type of cryptography, you can use only a single key. The sender and the receiver can use that single key to encrypt and decrypt a message.

Asymmetric key cryptography this cryptography differs from and is more secure than symmetric key cryptography. In this system, each user encrypts and decrypts using two keys or a pair of keys (private key and public key).

Each user keeps the private key secret and the public key is distributed across the network so that anyone can use those public keys to send a message to any other user. You can use any of those keys to encrypt the message and can use the remaining key for decryption. An RSA algorithm is an example of asymmetric key cryptography.

i. **Public Key** – This key is accessible to all over the internet and used to cipher the secret text.

ii. **Private Key** – As clear form the name it is a receiver's private key and only the receiver will be able use this key to decrypt the required message.

Hash Function This algorithm makes no use of any keys. A hash value with a fixed length is calculated based on the plain text, making it impossible to recover the plain text's contents.

Need of Hybrid Cryptography

Hybrid Cryptography techniques are prevents unauthorized party from discovering the content of communication. We know that classic encryption schemes have long been used for security purposes and they have been successful to some degree. So why use hybrid system. The basic answer is security. Basically, what hybrid cryptography does is that it mixes the effectiveness of symmetric encryption with easiness of public key encryption. It enhances security level and both types of encryption efficiency. It has many advantages. Firstly, it enables user to communicate through hybrid cryptography. Some may think asymmetric scheme will hinder speed of encryption. Simultaneously using symmetric scheme, it increases efficiency of system as well as both schemes. It provides better security and increased performance.

1.2. Statement of Problem

In the cloud environment, resources are shared among all of the users and individual. So it is difficult for the cloud provider to ensure security. As a result, it is easy for an intruder to access, misuse and destroy the original forms of data. The main problems of the existing cloud are file encryption and decryption AES key management, there is not enough security because this only use of AES algorithm for file encryption. Hence, a file security approach is achieved on cloud that should be good balance between speed and security. The approach

uses hybrid cryptosystem that is the key management problem solved by RSA algorithm, encrypted file store on server by a good performance of AES algorithm, the encryption key (AES key) are encrypted by public key of RSA algorithm finally encrypted AES key are stored on database and user verification using RSA algorithm, its optimize security issue.

1.3. Objectives

The objectives for working on this subject are discussed under the following categories in a general and specific ways.

1.3.1. General Objective

The general objective of this research work is to optimize the security of data on cloud computing using Hybrid Cryptography (AES, RSA and SHA-512) algorithm.

1.3.2. Specific Objectives

To achieve the general objective of the thesis, the following specific objectives are identified:

- To analyze the security of authentication and authorization issue on data storage.
- To implement authorization services using RSA algorithm.
- To implement authentication services using AES and SHA 512 algorithm.
- To evaluate the performance and security of the developed system using appropriate metrics.

1.4. Scope of the Study

The scope of my study only focuses on achieving to Secured file storage on cloud. This improving the organization data storage and the data access only authorized users of the organization.

1.5. Research Methodology

While dealing with the study of this thesis, we try to consider the following methodology:

1. File Upload/Encryption Process
    - Maximum file size30MB.
    - File type:
        - Document file(Word)
        - Power Point file

- Video and audio (mp3,mp4)
- Image File
- (jpg, png, gif, jpeg, pgp)
- PDF File
- Text File

➢ Data owner select file.

➢ Upload selected file.

➢ Restore access key from server.

➢ Store the access key with file id on database.

2. File Download/Decryption Process

➢ Select the download file.

➢ Restore access key from database

➢ Upload (Insert) access key on server

➢ Decrypt the key using RSA private key

➢ Decrypt the file using the secret key(AES)

➢ Finally download original file.

3. Tools

➢ Operating System: Ubuntu 18.04.4 LTS Server.

➢ Web server: Nginx

➢ Cloud Platform: Heroku

➢ Database: MySQL.

➢ IDE: PyCharm.

➢ Programming Language: Python.

➢ Performance metrics: CloudSim, Prometheus

1.6. Expected Outcome of the Project

The expected outcome of the research is to develop a new system which is highly secured and the storage data only access authenticated and authorized user.

1.7. Importance of the Proposed Project

The important of this , are used to protect information of against unauthorized access use of information, practically strong and infeasible to get attacked.

1.8. Contribution of the Research

This research showed that it's possible to increase the cloud data storage system by giving much

More focus on the authentication and authorization by using hybrid algorithms.

1.9. Research Question

To develop a technique based on this approach the following research questions are first formulated Model:

1 How can exist security techniques be used to develop a solution to optimize security of data on cloud using Hybrid cryptography techniques?

2 How security of data is ensured during access and sharing by authorized users?

3 How data integrity is verifiable by authorized users?

4 How only authorize user can be access private data?

1.10. Discussion

There is no doubt in the efficiency and performance of CC services and application. This field is used to improve applications and services in numerous organizations. Storing data on the cloud is easy to access for users but at the same time, it creates issues like confidentiality, integrity and availability. The technique of the proposed scheme proves the identity of the users authorized without the need to reveal their passwords.

The increase the encryption data of users and data owners by merging secret key and private key when the user accesses the data in the cloud. Some commonly techniques used for data security and privacy purposes are hybrid cryptography by combining AES and RSA algorithm. The can consider that RSA and AES algorithms are used to prevent the application from data loss attacks and it can be applied on top of all the internet protocols that are based

on the examinations recoded in this paper showing that all the algorithms are functioned well with different execution time and memory consumption. The find new solution for threats and attacks over cloud computing to provide the security and privacy of the cloud and its users. The discuss a number of possible security measures, which should be considered in any cloud-based model. Also, the cover some security functions, which should be in a model as a guarantee for safety checks of a system and its users and the protection of private and sensitive data for all users of the cloud. The system can solve many attacks such as replay attack and insider attacks, Moreover; the work enjoys many features as follows:

- It supports mutual authentication between authentication cloud server and a user.
- It offers user privacy; in our system, the password is saved in the service provider. Thus, the confirmation cost of the security-sensitive table is decreased.

To conclude the study of the proposed system overcomes the balancing between security and privacy according to client demand. It was implementing the proposed system by using hybrid cryptography to overcome the security risks. This thesis work uses multiple encryption techniques Using RSA algorithm will resolve issues like security and privacy. Cloud performance will also be better. The other proposed systems face main challenges like resource allocations, time issue and security. The only solution is to combining two algorithms so that resources can be utilized effectively. In this way, availability, integrity and confidentiality will be achieved easily. The security and privacy are the major concerns in user, data owner and Cloud for cloud operations, as well as for the different data which run with the cloud. These issues considered the performance of the Cloud and applications supported by them.

CHAPTER TWO
LITERATURE SURVEY

2.1. Introduction

Cloud computing security and privacy is the most widely discussed area in both industry and academic researchers. Therefore, in this section, a number of related works will be addressed and discussed as follows:

2.2. Literature Review

Chetan Gaikwad, 2018, In this paper the authors deal with the Open Stack software for creating cloud. System will use AES algorithm to encrypt the data at the time when storing data in the database. Thus provides less security, because to maintain the security of data/information in cloud storage system provides an encryption mechanism using only AES algorithm.

V.S. Mahalle, 2017, has presented the hybrid cryptography using AES and RSA algorithm. AES algorithm requires a single key. In hybrid algorithm three keys are used. For data upload on cloud mandatory keys are AES secret key and RSA public key. Private Key of RSA and AES secret key are essential to download data from cloud. Whenever use makes an effort to upload data on cloud first that file stored onto directory for short time. In encryption process first AES algorithm is applied on file after that RSA algorithm is applied on encrypted data. Reverse process is followed for decryption. After applying keys that file covert into encoded form and stored on cloud server. Advantages of hybrid algorithm are data integrity, security, confidentiality and availability. Disadvantage of RSA algorithm is large amount time essential for data encode and decode.

S. Garg, 2019, in this paper three algorithms are used for implementation of hybrid algorithm. User authentication purpose digital signature is used. Blowfish algorithm is used to produce high data confidentiality. It is symmetric algorithm. It uses single key. Blowfish algorithm need least amount of time for encode and decode. Sub key array concept is used into blowfish algorithm. It is block level encryption algorithm. The main aim of this hybrid algorithm is achieve high security to data for upload and download from cloud. Hybrid algorithm solves the security, confidentiality and authentication issues of cloud. The drawback of these papers is RSA algorithm is large amount time essential for data encryption and decryption.

V. Maitri, 2016, discussed Cloud storage issues are solved using cryptography and

steganography techniques. Block wise Data security is achieved using AES, RC6, Blowfish and BRA algorithms. Key information security is accomplished using LSB technique. Data integrity is accomplished using SHA1 hash algorithm. Low delay parameter is achieved using multithreading technique. With the help of proposed security mechanism data integrity, high security, low delay, authentication and confidentiality parameters are accomplished. The drawback of this system is only uses symmetrical algorithm but high level security uses hybrid system.

A. Mohtaetal, 2017 Has presented away to implement Third Party Auditor (TPA) who not only checks their liability of Cloud Service Provider (CSP) but also checks the consistency and accountability of data. The author has also discussed the challenges of open issue of integrity and data dynamics. The main objective is to solve the problem of data privacy, accountability and integrity of data. The proposed model involves the clients, cloud service provider and TPA. The suggested scheme involves retrieval of file, encryption and decryption of file, integrity checking from CSP and procedure for giving control to TPA. The responsibility of the TPA is to audit the cloud data storage efficiently. CSP provides service to client on successful authentication. Files are encrypted and decrypted using RSA algorithm. Message digest is generated using SHA-256 algorithm. After performing the desired file operation, clients send the data to CSP and TPA. The proposed scheme maintains the consistency at cloud data storage for CSP and client and also checks the integrity of data. The drawback of these papers is RSA algorithm is large amount time essential for data encryption and decryption.

A study in (Kumari, Chandini, Gagandeep Singh, Gursharan Singh, and Ranbir Singh Batth 2019) state that various security models have been suggested for enhanced security, there is still no universal solution for coping with both safety and privacy issues. CC poses various threats to privacy and security; namely loss of data and accessibility, account hijacking, user, and service provider problems. Further, it emphasize that safety systems should be paid more attention. Besides, the technique for the classification of knowledge to be followed depends on whether you need to defend yourself from outsider attackers or even insiders. Current ways of dealing with insiders safely rely on protocols for location abnormality.

(Deorankar, A. and K.T. Khobragade 2020) paper addresses the problems of consumer safety; attention is also given to the competitiveness of the proposed system. The authors propose that theclient should get nothing into the database, and the server should not collect any information about the biometric relation. The property-based encryption method was proposed for the identification of sophisticated sharing of data. The fine-scale, high data consumer performance, and standard privacy of cloud data sharing still remain unclear.

(According to Fatima, S. and S. Ahmad 2019) security and privacy are the critical challenges of the computing application. This paper identifies several aspects of protection, including confidentiality, availability, and honesty. It also debates on protection and privacy issues. According to this paper, data encryption is the safest possible way to maintain confidentiality. However, essential control is a severe problem of encryption. The data owners cannot presume to handle keys. Key management is expected from cloud service providers, and this is not a good option. The object of data integrity is to protect data from unlawful users being altered or deleted. In centralized structures, data confidentiality can be easily achieved. If unauthorized users deduce something from the cloud data, the privacy of the data is impaired. Misuse of data and the prevention of attacks are the most significant privacy risks to cloud storage. The attacker aim to misuse and destroy cloud data and will try to access it using the hash code. This causes the confidential data to be leaked. Different cloud services providers, including Microsoft Azure, Google Web Services, etc. provide cost-effective cloud access. They offer consumer and customer services on demand that pays what they get but security, privacy, and data protection are still significant concerns. It involves the production and distribution of replicas of shares among different providers of resources. For encrypting the shares, the AES algorithm is employed. The scheme and AES algorithms created shares are known as encapsulated shares. The technique is proposed to improve safety and reduce fraud. The technology offered is used to address the complex maintenance issue.

M. M. A Hashem. 2019, In these paper the researchers have proposed new security architecture for exchanging information in a cloud computing environment. In order to ensure a secure communication process, the proposed architecture includes AES file encryption system, RSA system for secure communication, onetime password to authenticate users and MD5 hashing for hiding information. The main objective behind the proposed model is to solve main security issues like malicious intruders, hacking, etc. in cloud computing platform.

Additionally, distributive server concept is used, thereby ensuring higher security for the whole cloud. Experimental results have also been shown to determine the efficiency of the proposed model. From the comparative analysis, it is demonstrated that the proposed model works smoothly and ensures higher security than other present running models in a cloud computing environment.

Though the proposed architecture is highly secure, but RSA encryption system becomes fragile in long run process. Thus, there is a need to find out a light and secure encryption system for secure communication and hiding information from others.

D.Mukhopadhyay, 2018 proposed a methodology to overcome security threats which emerge with shared spaces in the cloud environment. The proposed framework involves securing of files through file encryption mechanism. The file present on the device is encrypted using password based AES algorithm. The user can download any of the uploaded encrypted files and read it on the system aft the successful decryption. The suggested scheme aims at preventing every possible attack on the user data existing on the cloud. The integrity and confidentiality of the data uploaded by the user is ensured doubly by not only encrypting it but also providing access to the data only on successful authentication. The authors have implemented the proposed framework to encrypt only text files which can be further enhanced to encrypt the audio and video files.

M Abhishek Sachdeva, 2013 deals with comparison between AES, DES and 3DES. Advanced Encryption Standard (AES) algorithm is an encryption algorithm to maintain data confidentiality. Both hardware and software implementations are faster still. It is the new encryption standard recommended by NIST to replace Data Encryption Standard Algorithm (DES). According to, AES is more secure and efficient compared to DES and 3DES. Research results in comparison between AES, DES and 3DES as shown in the Table 2.1.

Table 2. 1: Comparison between AES, DES and 3DES

Factors	**AES**	**3DES**	**DES**
Key Length	128, 192, or 256 bits	(k1, k2 and k3) 168 bits (k1 and k2 is same) 112 Bits	56 bits
Cipher Type	Symmetric block cipher	Symmetric block Cipher	Symmetric block cipher

Other research result why AES is preferable over the others is stated as follows: Why AES?

Block Size	128 bits	64 bits	64 bits
Developed	2000	1978	1977
Cryptanalysis resistance	Strong against differential, truncated differential, linear, interpolation and square attack	Vulnerable to differential, Brute Force attacker could be analyzed a plain text using differential cryptanalysis.	Vulnerable to differential and cryptanalysis; weak substitution tables
Security	Considered secured	One only weak which is Exit in DES	Proven inadequate
Possible key	2^{128}, 2^{192} or 2^{256}	2^{112} or 2^{168}	2^{56}
Time required to check all possible at 50 billion keys per second	For a 128-bit key: $5*10^{21}$ years	For a 112-bit key: 800 days	For a 56-bit key: 400 days

- AES performs consistently well in both hardware and software platforms under a wide range of environments. These include 8-bit and 64-bit platforms.
- Its inherent parallelism facilitates efficient use of processor resources resulting in very good software performance.
- This algorithm has speedy key setup time and good key agility.
- It requires less memory for implementation, making it suitable for restricted-space environments.
- The structure has good potential for benefiting from instruction-level parallelism.
- There are no serious weak keys in the AES.
- It supports any blocks size and key sizes that are multiples of 32(greater than 128-bits).

- Statistical analysis of the cipher text has not been possible even after using a huge number of test cases.
- No differential and linear cryptanalysis attacks have been yet proved on AES.

A performance comparison among AES, DES and triple DES for different microcontrollers shows that AES has a computational cost of the same order as required for triple DES. Another performance evaluation reveals that AES has an advantage over algorithms Triple Data Encryption Standard (3DES), DES and Rivest Cipher 2 (RC2) in terms of execution time with different packet size and throughput for encryption as well as decryption. Also in the case of changing data type such as image instead of text, it has been found that AES has an advantage over RC2, Rivest Cipher 6 (RC6) and blowfish in terms of time consumption.

2.3. Related Works

Comparison of Related works in security and privacy concern for this project is summarized in table.

Table 2. 2: Related works in security and privacy concern

Paper Title	Algorithm Used	Strong Points	Weak Points
Chetan Gaikwad.	AES(file encrypting/ decryption)	Storing user's personal data on private cloud using OpenStack software only authorized user will get the access data.	AES key management and Uses only AES algorithm.
V.S.Mahalle	AES and RSA	Assured data integrity, security, confidentiality and availability.	RSA algorithm is large amount time essential for data encryption and decryption.
S. Garg	RSA as Digital Signature is used for authentication and non-repudiation purpose Blowfish	Assured data integrity, security, confidentiality authenticity and non-repudiation	RSA algorithm is large amount time essential for data encryption and decryption.
V. Maitri	AES,RC6,Blowfish	Provide block wise security to data.	AES,RC6,Blowfish providing low delay for data encode decode but provides low security.
M.M.A Hashem.	RSA,AESand MD5	Overcome the security trade- off and improve the performance of data transmission and increase the Security	MD5 hashes are no longer consider cryptography secure
A.Mohtaetal	RSA and AES	Minimizing the consumption of time, cost and memory size during encryption and decryption.	RSA is a computationally costly. AES is better than RSA in terms of time complexity but not more useful to distributing key

In general, the reviewed researcher used different techniques and cryptography algorithm for secure data storage on the cloud computing.

CHAPTER THREE
METHODOLOGY

3.1. Introduction

In this section of the study, we are going to describe how the proposed system will work and how different techniques are used. This section includes the description about methods that can help for secure file storage system on cloud, give us a direction for implementing an AES algorithm, One Time Password and RSA algorithm and also proposes new method to secure file storage system between user and cloud by using hybrid cryptosystem.

3.2. Hybrid Cryptosystem

In order to ensure file security on cloud, hybrid cryptosystem is being used. We assume that the remote server is trusted, so files are encrypted by server, finally encrypted files are stored at the server end and the encrypted secret key (AES key) are stored at the database. The proposed hybrid cryptosystem involves the following algorithms:

- AES Algorithm: used for file encrypted.
- RSA Algorithm: used for digital signature.

3.2.1. AES Algorithm

According to our achieved knowledge in chapter 2, AES gives us the most secure algorithm for data encryption/decryption, so we will use AES in our system. It is necessary to use a key length, which must at least be 128 bits long to achieve the necessary security. A key length of more than 128 bits would decrease the performance of encryption.

The AES algorithm operates on 128 bits of data and generates 128 bits of output. The length of the key used to encrypt this input data can be 128, 192 or 256 bits. As this is a symmetric key cipher it uses the same key for both Encryption and Decryption. Nb which defines the number of columns of 32 bits is, Nb =128/32=4. Similarly, Nk which defines the number of columns of 32 bits of key is, Nk =128/32= 4. For key length of 192 and 256 the values of Nk will be 192/32= 6 and 256/32= 8 respectively. The number of rounds Nr =10 when Nk= 4 and changes to 12 and 14 for Nk =6 and Nk =8 respectively. The key block round combinations are given in Table.3.1. As the key length is 128 bits i.e. Nk = 4, Nr will be 10.

Table 3. 1: Key Block Round Combination

	Key Length	No. of columns (Nk)	No. of rounds (Nr)
AES-128	128 bit	4	10
AES-192	192 bit	6	12
AES-256	256 bit	8	12

The AES algorithm basically consists of four-byte oriented transformation and a key expansion function. The transformations are repeated for 10 rounds by applying the inputs to produce cipher text. For the first nine rounds all four blocks are repeated but for the final round the Mix Columns transformations is excluded.

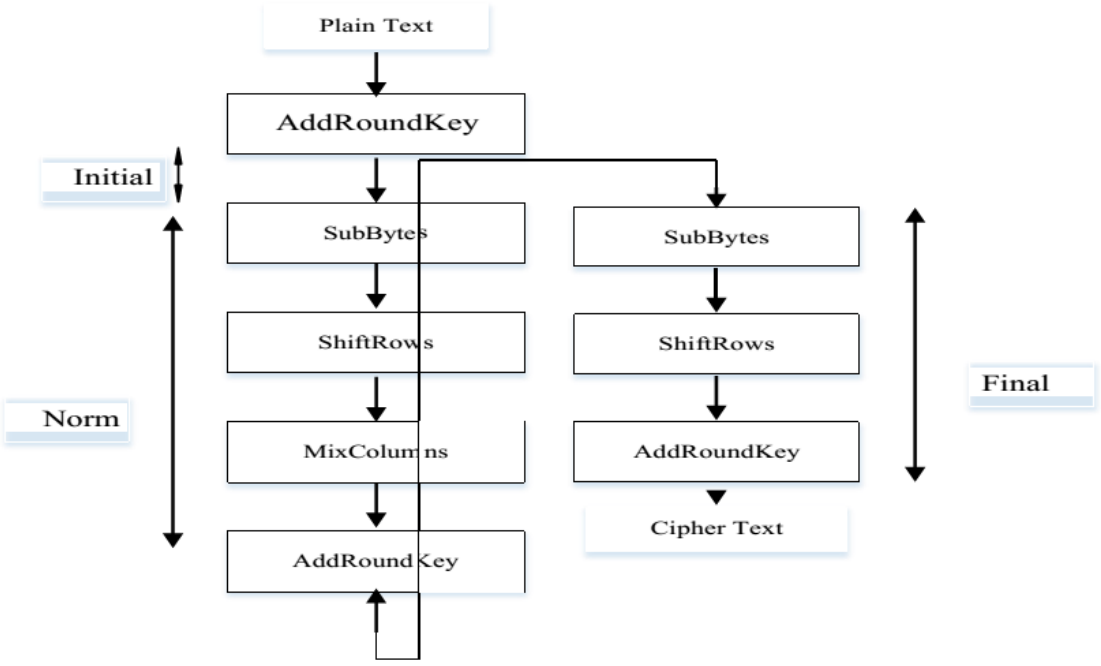


Figure 3. 1: Basic building block of AES

The basic building block of AES containing four separate blocks, Sub Bytes, Shift Rows, Mix Columns and Add Round Key are shown in Figure 3.1.

The Add Round Key transformation involves a bitwise XOR operation between the state array and the resulting Round Key, which is derived from the initial key in the Key Expansion function. Sub Bytes transformation is a highly non-linear byte substitution where each by the in the state array is replaced with another from a lookup table called an S-Box.

3.2.1.1. Key Scheduling

AES uses a key schedule to expand a short key into a number of separate round keys. The three AES variants have a different number of rounds. Each variant requires a separate 128-bit round key for each round plus one more. The key schedule produces the needed round keys from the initial key.

Round constants

The round constant *r con*$_i$ for round *i* of the key expansion is the 32-bit word:

Table 3. 2: Values of rci in hexadecimal

I	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Rci	**01**	**02**	**04**	**08**	**10**	**20**	**40**	**80**	**1B**	**36**

rcon$_i$ = [**rc*i* 00*16* 00*16* 00*16***]

Where *rc*$_i$ is an eight-bit value defined as:

$$rc_i = \begin{cases} 1 & \text{If i=1} \\ 2 .* r & \text{If i=1 and } rc_{i-1} < 80_{16} \\ (2 * rc_{i-1}) \oplus 1B_{16} & \text{If i=1 and } rc_{i-1} \geq 80_{16} \end{cases}$$

Where o is the bitwise XOR operator and constants such as $00_{16}$and$1B_{16}$ are given in hexadecimal. Equivalently:

$rc_i = \mathbf{x^{i-1}}$

Where the bits of *r c*$_i$ are treated as the coefficients of an element of the finite field GF (2) [x]($x^8 + x^4 + x^3 + x + 1$), so that e.g. r $c_{10}=36_{16}=00110110_2$ represents the polynomial $x^5 + x^4 + x^2 + x$.

AES uses up to *r con*10 for AES-128 (as 11 round keys are needed), up to *rcon*8 for AES-192, and up to *rcon*7 for AES-256.

Key schedule

Define:

- *N* as the length of the key in 32-bit words: 4 words for AES-128, 6 words for AES-192, and 8 words forAES-256
- $K_0, K_1, \ldots K_{N-1}$ as the 32-bit words of the original key
- *R* as the number of round keys needed: 11 round keys for AES-128, 13 keys forAES-192, and 15 keys forAES-256
- $W_0, W_1, \ldots W_{4R-1}$ as the 32-bit words of the expanded key Also define Rot Word as a one-byte left circular shift:

Rot Word ([b0 b1 b2 b3]) = [b0 b1 b2 b3] and Sub Word as an application of the AES S-box to each of the four bytes of the word:

Sub Word ([b0 b1 b2 b3]) = [S(b0) S(b1) S(b2) S(b3)] Then for i= 0 . . . 4R-1

$$W = \begin{cases} W_i & \text{If } I < N \\ W_{i-N} + \text{Sub Word}(\text{Rot Word}(W_{i-N})) + r\,con_{i/N} & \text{If } I \geq N \text{ and } I = (m \\ W_{i-N} + \text{Sub Word}(W_{i-1}) & \text{If } I \geq N, N>6, \text{ and } I = 4 \pmod N \\ W_{i-N} + W_{i-1} & \text{Otherwise.} \end{cases}$$

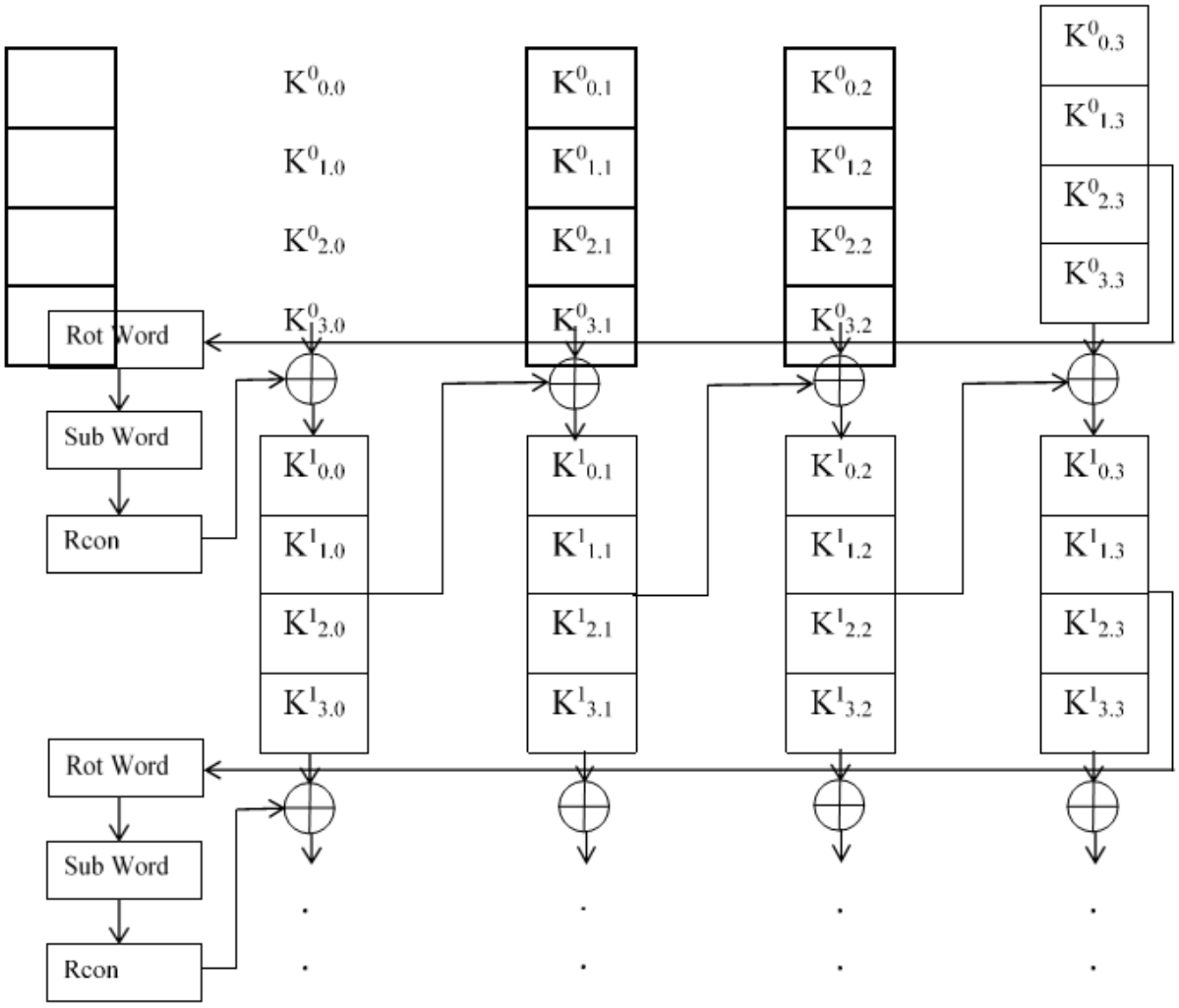


Figure 3. 2: AES key schedule for a 128-bit key

3.2.1.2. Data Encryption Process

Encryption is a popular technique that plays a major role to protect data from intruders. AES algorithm uses a particular structure to encrypt data to provide the best security. To do that it relies on a number of rounds and inside each round comprise off our sub-processes. Each round consists of the following four steps to encrypt 128-bit block.

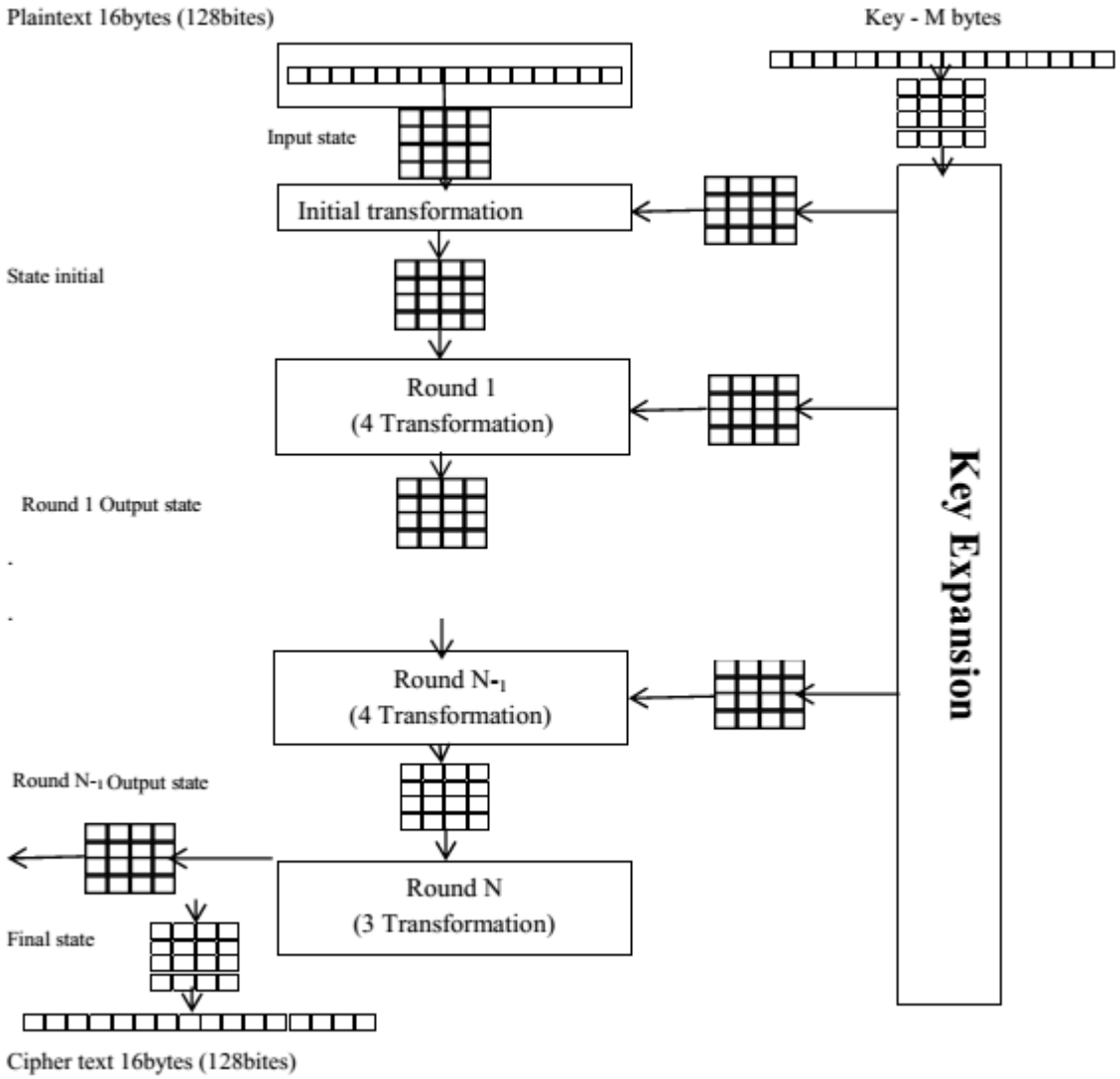


Figure 3. 3: AES data encryption processes

Substitute Bytes Transformation

The first stage of each round starts with Sub Bytes transformation. This stage is depending on nonlinear S-box to substitute a byte in the state to another byte. According to diffusion and confusion Shannon's principles for cryptographic algorithm design it has important roles to obtain much more security. For example, in AES if we have hexa 53 in the state, it has to replace to hexa ED. ED created from the intersection of 5 and 3. For remaining bytes of the state have to perform this operation.

Table 3. 3: AES S-box Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	5B	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	AB
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	EO	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	CB	37	6D	8D	D6	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	FB	0E	61	35	57	B9	86	C1	1D	9E
E	E7	F8	9B	11	69	D9	BE	94	9B	1E	87	E9	CE	55	28	DF
F	BA	A1	89	0D	BF	BF	42	6B	41	99	2D	0F	B0	54	BB	16

Shift Rows Transformation

The next step after SubByte that performs on the state is ShiftRow. The main idea behind this step is to shift bytes of the state cyclically to the left in each row rather than row number zero. In this process the bytes of row number zero remains and do not carry out any permutation. In the first row only one byte is shifted circular to left. The second row is shifted two bytes to the left. The last row is shifted three bytes to the left. The size of new state is not changed that remains as the same original size 16 bytes but shifted the position of the bytes in state as illustrated in Fig3.4.

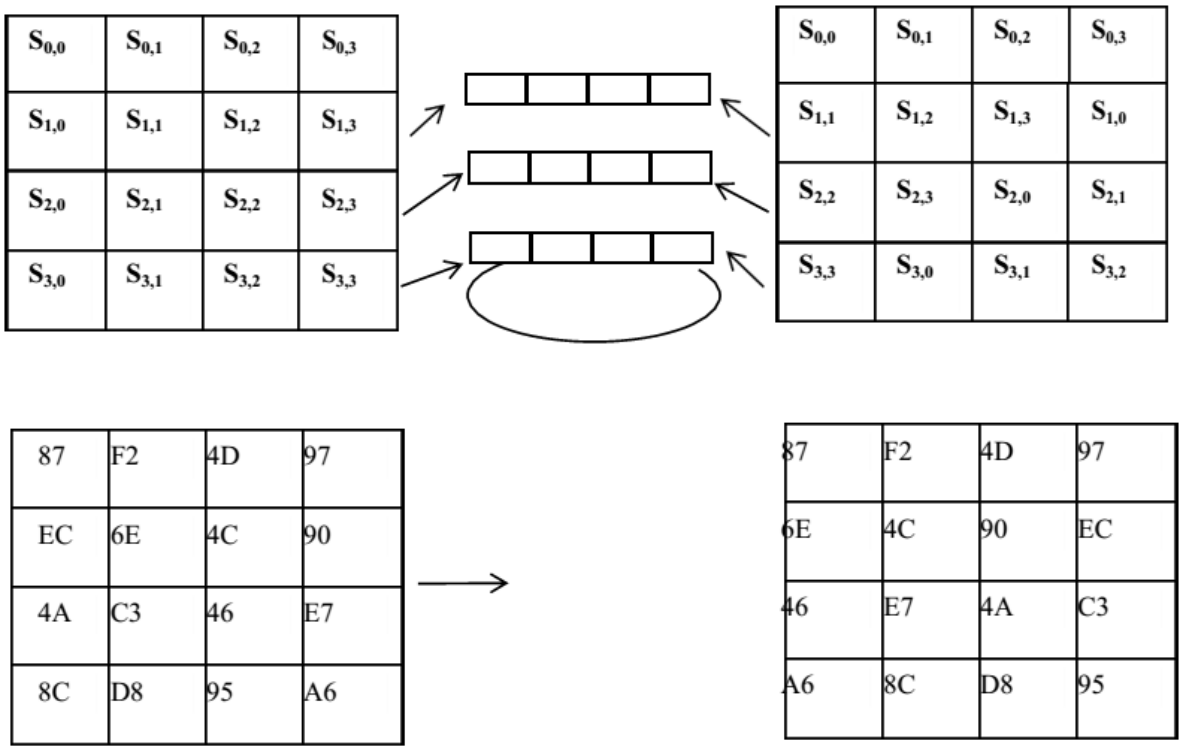


Figure 3. 4: Shift Rows

Figure 3. 4: Shift Rows

Mix Columns Transformation

Another crucial step occurs of the state is Mix Column. The multiplication is carried out of the state. Each byte of one row in matrix transformation multiplies by each value (byte) of the state column. In another word, each row of matrix transformation must multiply by each column of the state. The results of this multiplication are used with XOR to produce a new four bytes for the next state. In this step the size of state is not changed that remained as the original size 4x4 as shown in Fig.3.5.

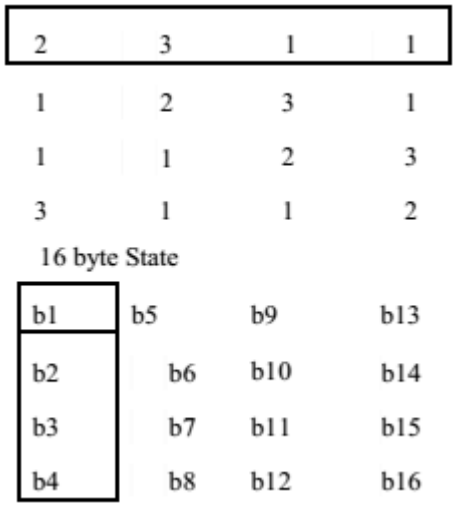


Figure 3. 5: Multiplication Matrix 36

Add Round Key Transformation

Add Round Key is the most vital stage in AES algorithm. Both the key and the input data (also referred to as the state) are structured in a 4x4 matrix of bytes. Fig3.6 shows how the 128-bit key and input data are distributed into the byte matrices. Add Round Key has the ability to provide much more security during encrypting data. This operation is based on creating the relationship between the key and the cipher text. The cipher text is coming from the previous stage. The Add Round Key output exactly relies on the key that is indicated by users. Furthermore, in the stage the sub key is also used and combined with state. The main key is used to derive the sub key in each round by using Rijndael's key schedule. The size of sub key and state is the same. The sub key is added by combining each byte of the state with the corresponding byte of the sub key using

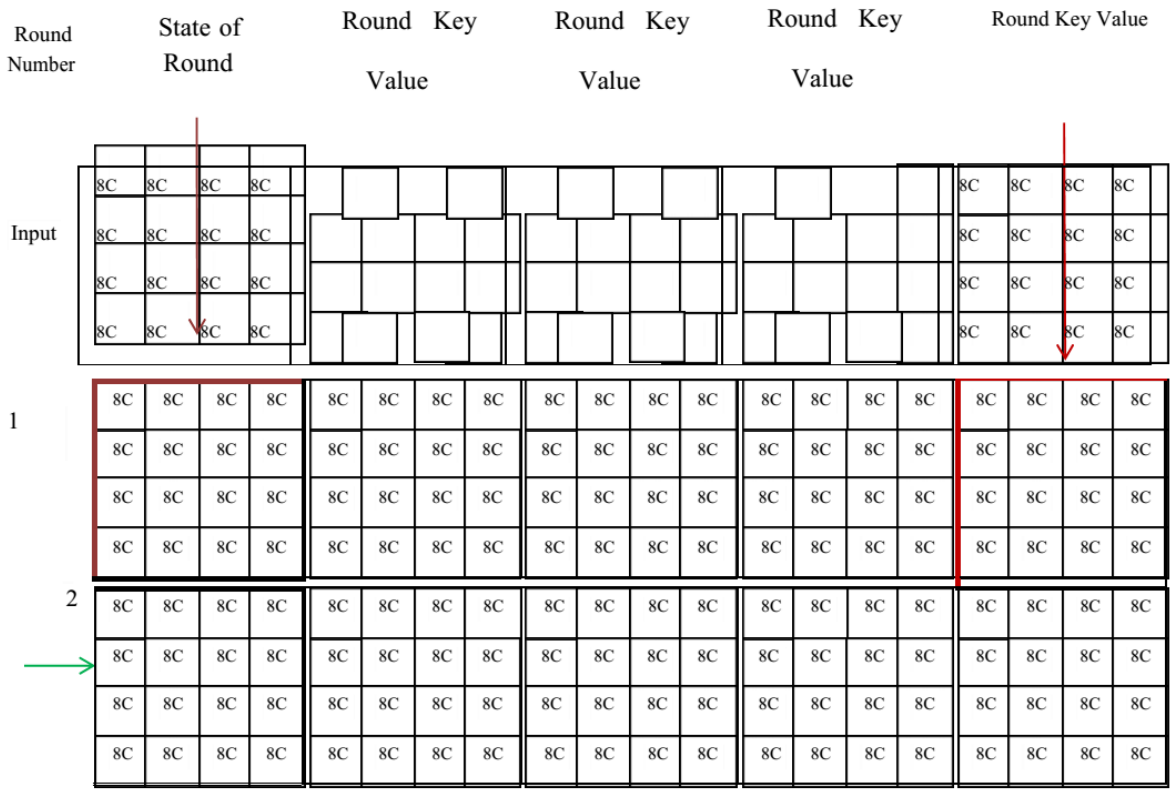


Figure 3. 6: Add Round Key

The decryption is the process to obtain the original data that was encrypted. This process is based on the key that was received from the sender of the data. The decryption processes of an AES are similar to the encryption process in the reverse order and both sender and receiver have the same key to encrypt and decrypt data. The last round of a decryption stage consists of three stages such as InvShiftRows, InvSubBytes, and Add Round Key as illustrated in Fig.3.7.

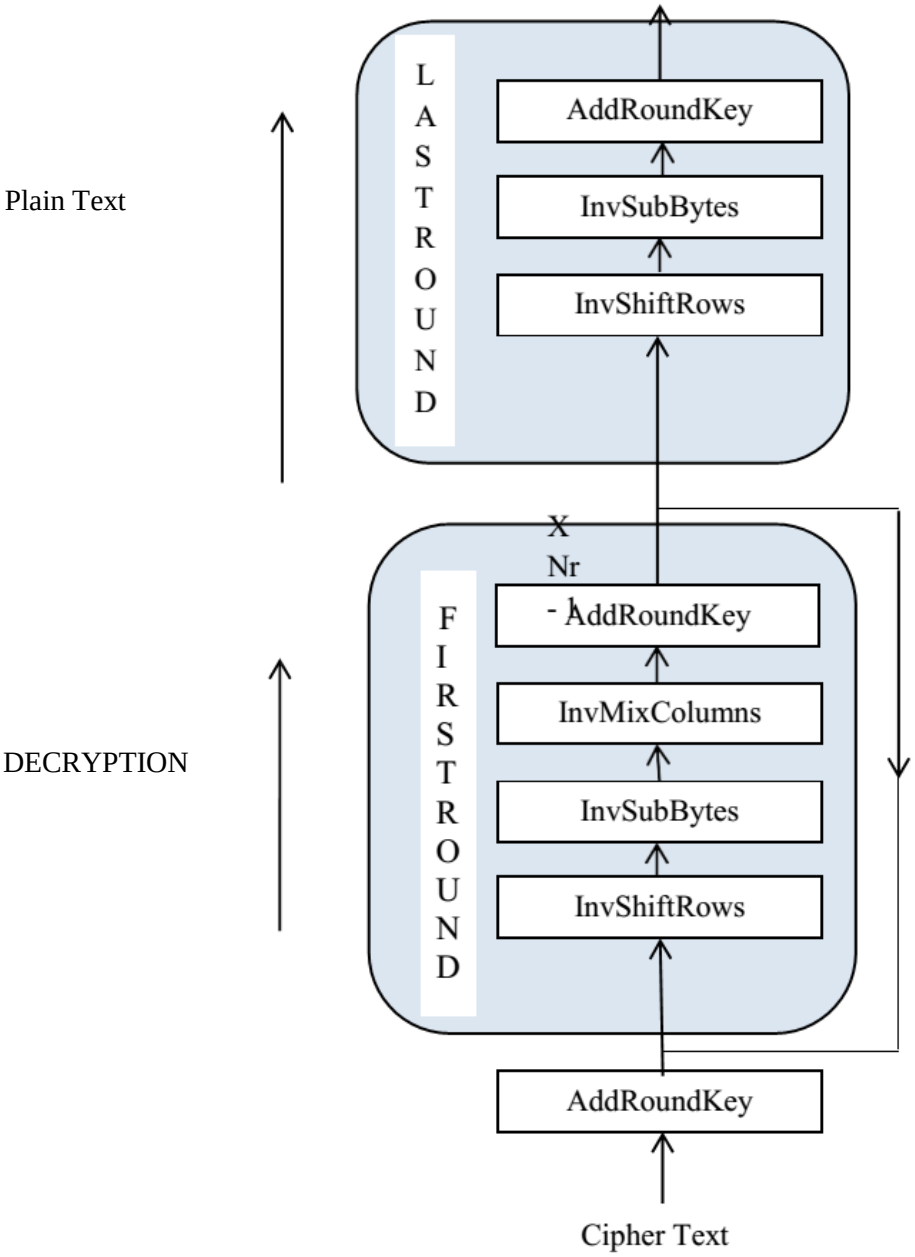


Figure 3. 7: Process of AES data decryption

3.2.2.RSA Algorithm

The RSA algorithm is an asymmetric cryptography algorithm; this means that it uses a public key and a private key (i.e two different, mathematically linked keys). As their names suggest, a public key is shared publicly, while a private key is secret and must not be shared with anyone.

The RSA algorithm is named after those who invented it in 1978: Ron Rivest, Adi Shamir, and Leonard Adleman.

3.2.2.1. Key Generation Process

The key generation process of RSA is as follows:

- Select two large prime numbers, x and y. The prime numbers need to be large so that they will be difficult for someone to Figure out.
- Calculate $n=x$ x y.
- Calculate the totient function; $\phi(n) = (x-1)(y-1)$.
- Select an integer e, such that e is ***co-prime*** to $\phi(n)$ and $1 < e < \phi(n)$. The pair of numbers (n, e) makes up the public key.
- Calculate d such that $e.d=1 \ mod \ \phi(n)$.

3.2.2.2. Encryption Process

Given a plaintext P, represented as a number, the ciphertext C is calculated as:

$C=P^e \ mod \ n$.

3.2.2.2 Decryption Process

Using the private key (n, d), the plaintext can be found using:

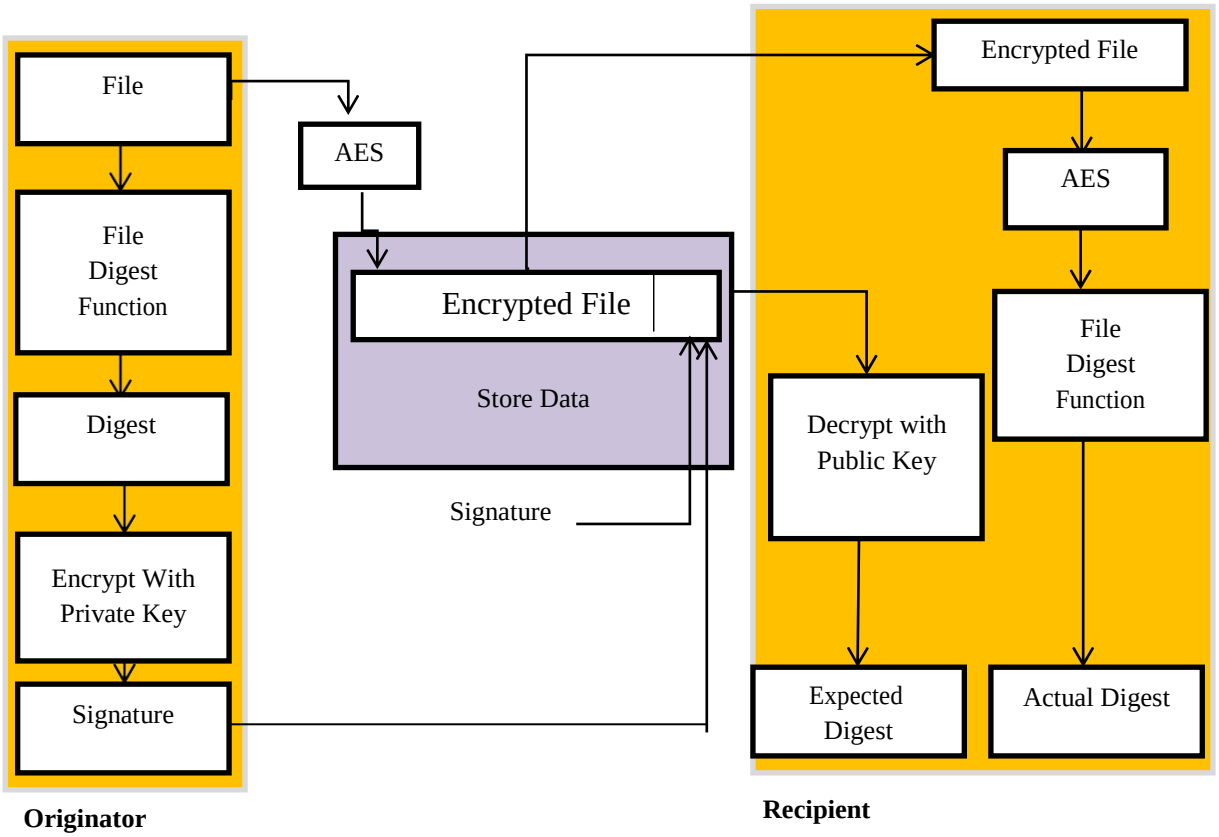
$P=C^d \ mod \ n$.

3.2.2.3. RSA Digital Signature Scheme

In the RSA digital signature process, the private key is used to encrypt the file digest. The encrypted file digest becomes the digital signature.

A digital signature scheme typically consists of three algorithms:

- A key generation algorithm that selects a private key uniformly at random from a set of possible private keys. The algorithm outputs the private key and a corresponding public key.
- A signing algorithm that produces a signature, when given a file digest and a private key.
- A signature verifying algorithm that either accepts or rejects the file 's claim to authenticity, when given a file, public key and a signature.



digest and

If the expected actual digest are the same, the signature is verified. 40

Figure 3. 8: Process of RSA digital signature algorithm

3.3. Proposed Algorithm

Figure 3.9 shows the algorithm flow diagram of the proposed system contains authentication feature in which user must register first with their details (username, email address and password) which is being stored in database and then login with their own user email address and password. If user details are stored in database, then only authorized user will have access the data. The proposed system provides efficient data handling and gives the user the privilege to add, delete and update the data anytime as per users need. Data which is uploaded are handled with the help of various Flask framework. In proposed system is basically a web application which includes login and registration like Gmail and soon.

This proposed system allows the user to store or retrieve data from the system. Whenever a user tries to upload a file, a AES key will be generated and that key will be used to encrypt the file. The encrypted file path is stored in the database, the encrypted files are stored in the server and AES key encrypted by public key of RSA algorithm and also store decrypted AES Key on database. Whenever a user tries to download the selected file the user needs to restore

Access key from database. Using this Access key and FileID, the secret key of that particular file can be decrypted private key of RSA algorithm in this key decrypt the file using AES algorithm by application server finally download the original file.

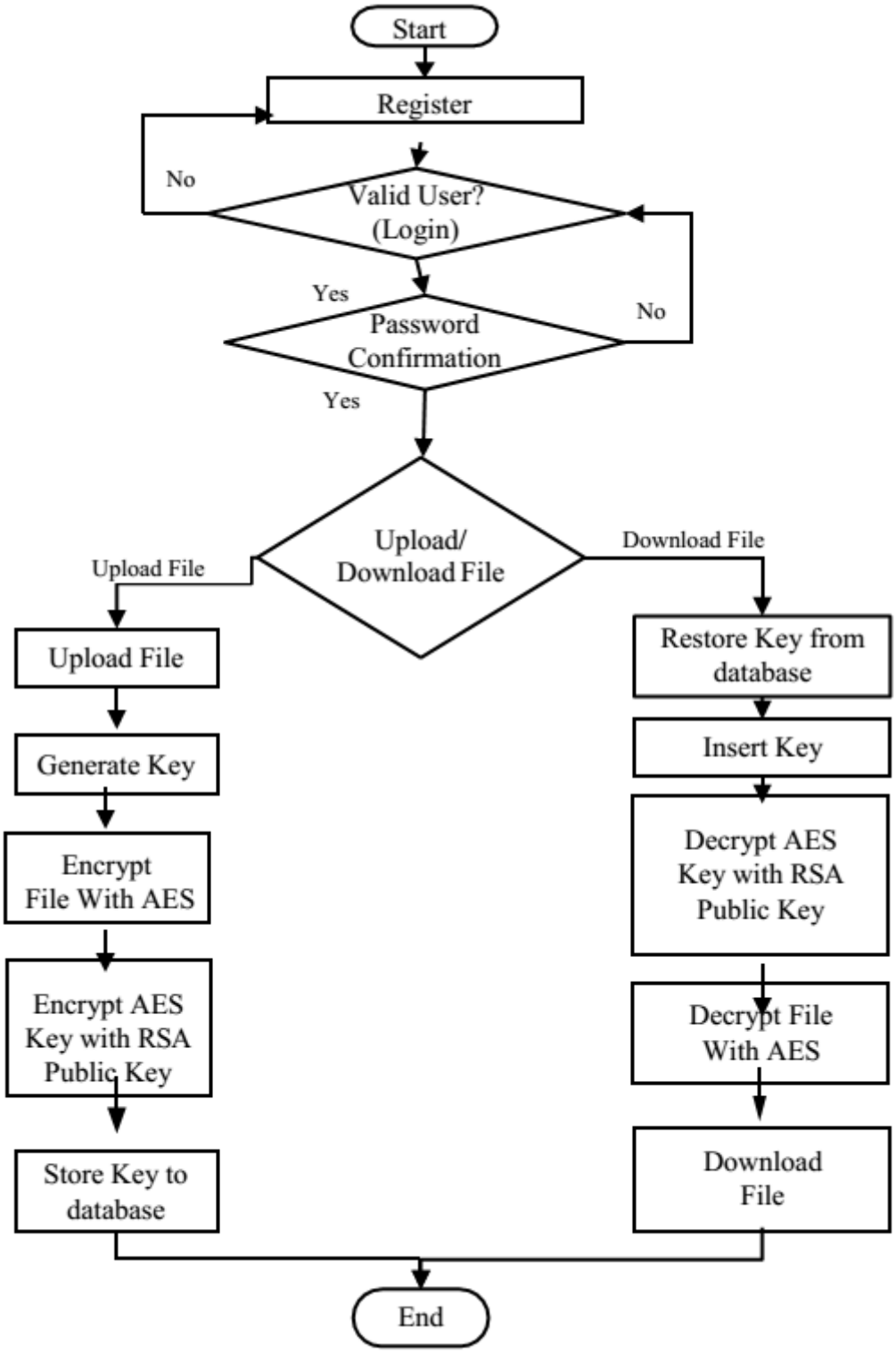


Figure 3. 9: Algorithm flow diagram of proposed system

3.4. Parts of the Proposed System

A Data Flow Diagram (DFD) is the graphical representation of the flow of data through a proposed system. DFD is very useful in understanding a system and can be efficiently used during analysis. A DFD shows the flow of data through a system. It views a system as a function that transforms the inputs into desired outputs. With a data flow diagram, users are able to visualize how the system will operate that the system will accomplish and how the system will be implemented.

Proposed system are mainly working in below three major part.

3.4.1. User Account Operations

a) Register a new user such as username, email address, and password.

b) Login to an existing account with email address, and password.

c) Data owner edit the existing Profile such as username, email address, password and profile picture.

d) Forgot Password and receive the current password over an email.

e) Logout from the session.

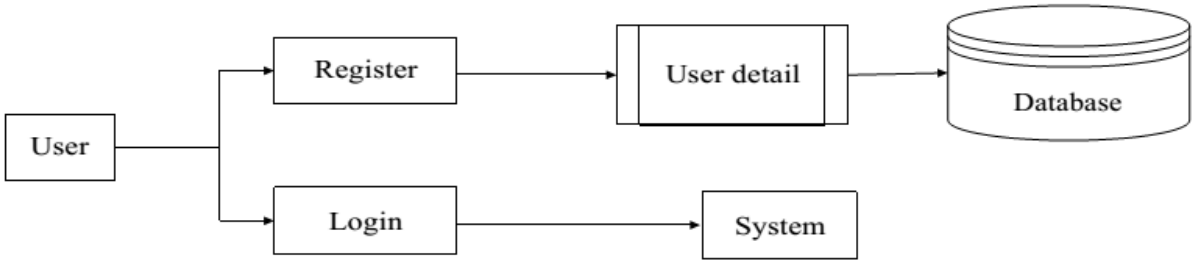


Figure 3. 10: Data flow diagram for user account

3.4.2. File Uploading Process

Whenever a user tries to upload public file can be directly whereas to upload a private/secret file the user needs secret AES key will be generated and that key will be used to encrypt the file. The RSA algorithm used to encrypt the AES key is stored in the local database and the encrypted data and digital signature are stored in the server.

f) Data owner upload the selected file on server.

g) Generating 128 bit AES key.

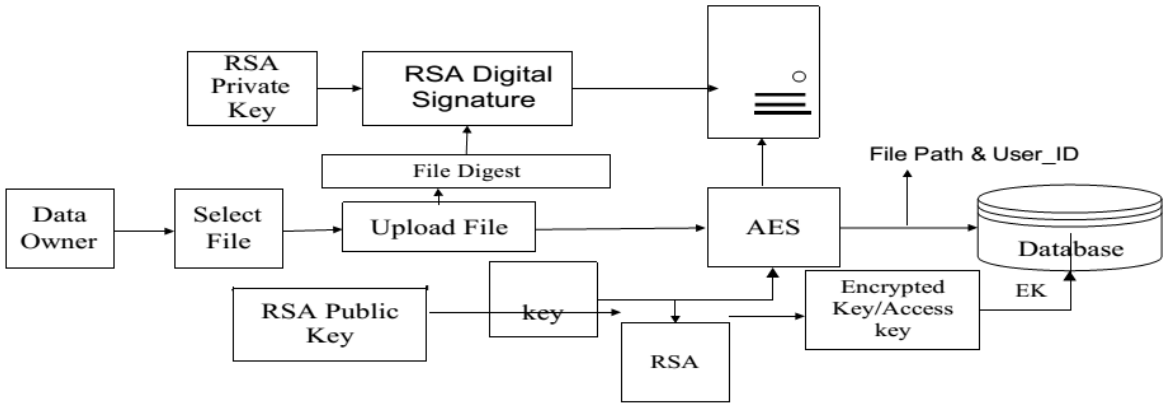
h) AES algorithm is used to encrypt upload file using cipher keys of 128 bits' length.

i) Generate Digital Signature (DS) using RSA algorithm.

j) RSA algorithm is used to encrypt AES key using public keys of 1024 bits' length.

k) Finally, Data owner store access key with fileID on database.

DS Server FH



EF = Encrypted File EK = Encrypted Key DS = Digital Signature FH = File Hash

Figure 3. 11: Data flow diagram for file uploading/encryption process

3.4.3. File Downloading Process

Whenever a user tries to retrieve the public file can be downloaded directly whereas to retrieve the private/secret files the user needs to request for an access key. Using this Access key for that particular file can be taken from the server and the file can be decrypted and downloaded.

l) Select the download file.

m) Restore Access Key from Database.

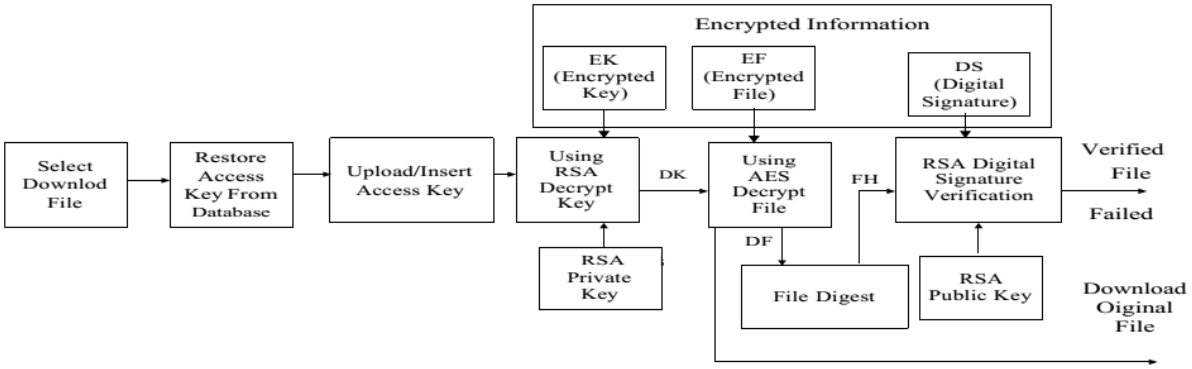
n) Upload/Insert the access key on a selected encrypted file.

o) Decrypt the keys using RSA algorithms.

p) Decrypt the file using the AES algorithms.

q) Verifying file using RSA Digital Signature algorithm.

r) Finally, data owner downloading original file.



DF = Decrypted File DK = Decrypted Key DS = Digital Signature DH = File Hash

Figure 3. 12: Data flow diagram for file downloading /decryption process

CHAPTER FOUR

DESIGN AND IMPLEMENTATION

4.1. Implemented Tools for Project

Proposed system is based on Flask framework, micro web framework of Python programming. Selection on Flask is my good choice as being a web-framework it allows us to move with fast development and keep open the space for future enhancements. It's based on Python Programming.

4.1.1. Back Side Language Python 3.6

- Python Libraries
- Pillow
- OS
- Sys
- Smtplib
- Datetime
- Werkzeu

4.1.2. Web Framework (Flask)

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, and upload handling, various open authentication technologies, and several common frameworks related tools. Extensions are updated far more regularly than the core Flask program.

Flask Features:

Flask was designed to be easy to use and extend. The idea behind Flask is to build a solid foundation for web applications of different complexity. From then on you are free to plug in any extensions you think you need. Also you are free to build your own modules. Flask is great for all kinds of projects. It's especially good for prototyping. Flask depends on two

external libraries: The Jinja2 template engine and the Werkzeug WSGI toolkit.

Still the question remains why use Flask as your web application framework if we have immensely powerful Django, Pyramid, and don't forget web mega framework Turbo gears? Those are supreme Python web frameworks BUT out-of-the-box Flask is pretty impressive too with its:

- built-in development server and fast debugger
- integrated support for unit testing
- RESTful request dispatching
- Jinja2 tinplating
- support for secure cookies (client-side sessions)
- WSGI 2.0.19.1 compliant
- Unicode based **Version Used in Application** Flask==1.1.6

4.1.3. Front End Design

The frontend of a website is what you see and interact with on your browser. Also referred to as "client-side", it includes everything the user experiences directly: from text and colors to buttons, images, and navigation menus.

To make easy access of the application we use easy to navigate screen and upload and restore file button, with user friendly and self-understood interface.

- HTML
- CSS
- JavaScript (JS)
- Bootstrap
- MYSQL Server
- Nginx Web-server

4.1.4. Development Tools Pycharm

PyCharm is an integrated development environment used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains.

4.2. Proposed System Screenshots

4.2.1. Registration and Login Process

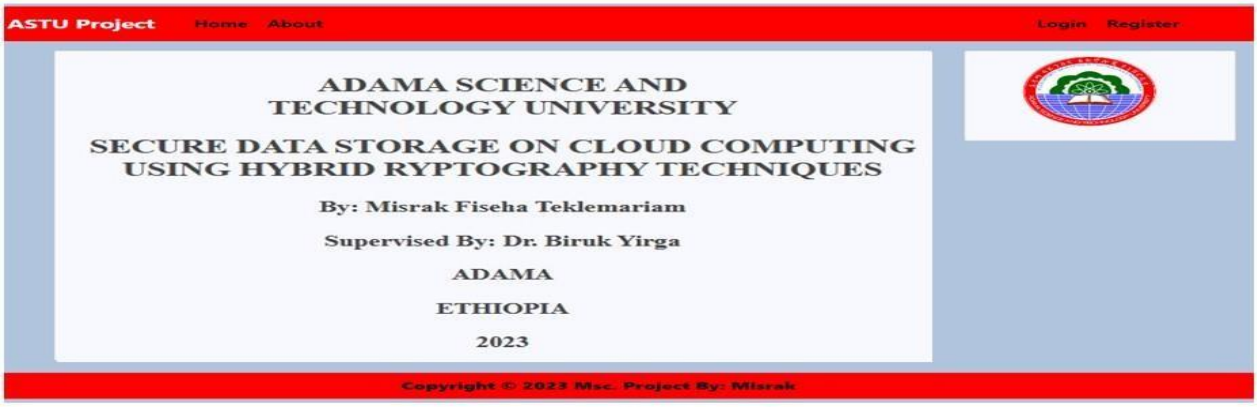


Figure 4. 1: Home page

Figure 4. 2: User registration form

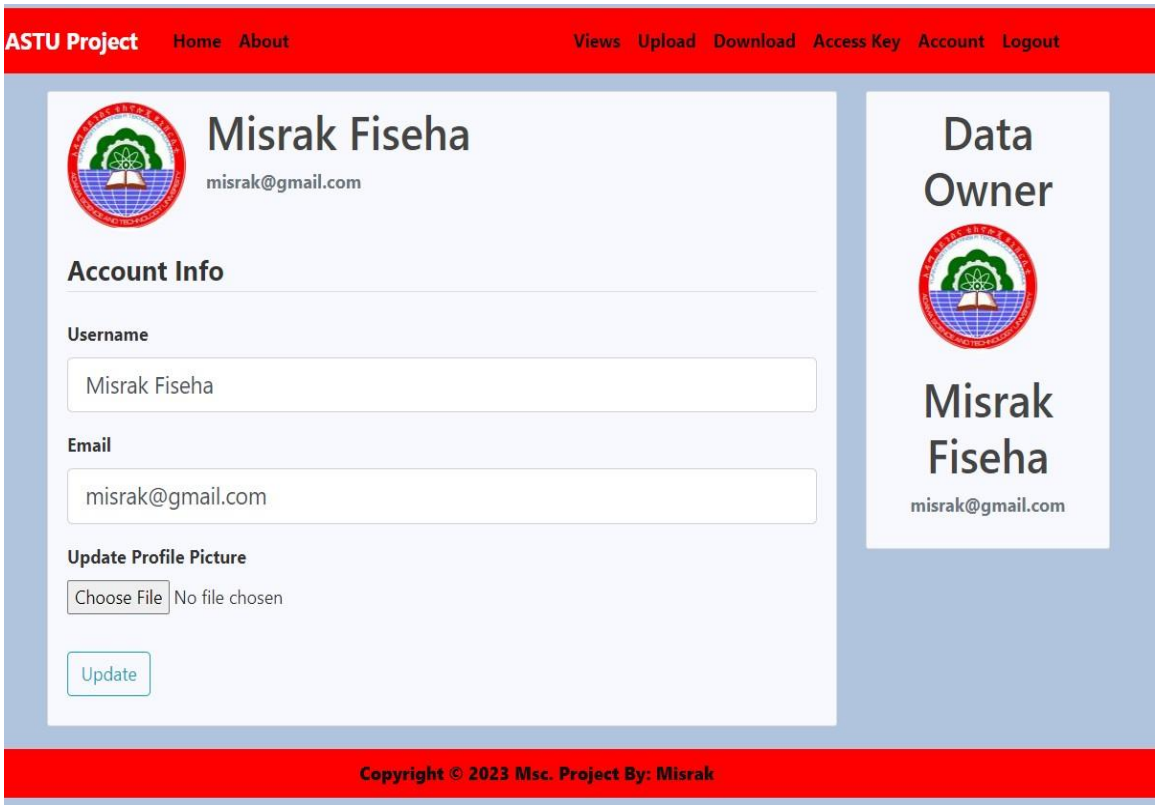
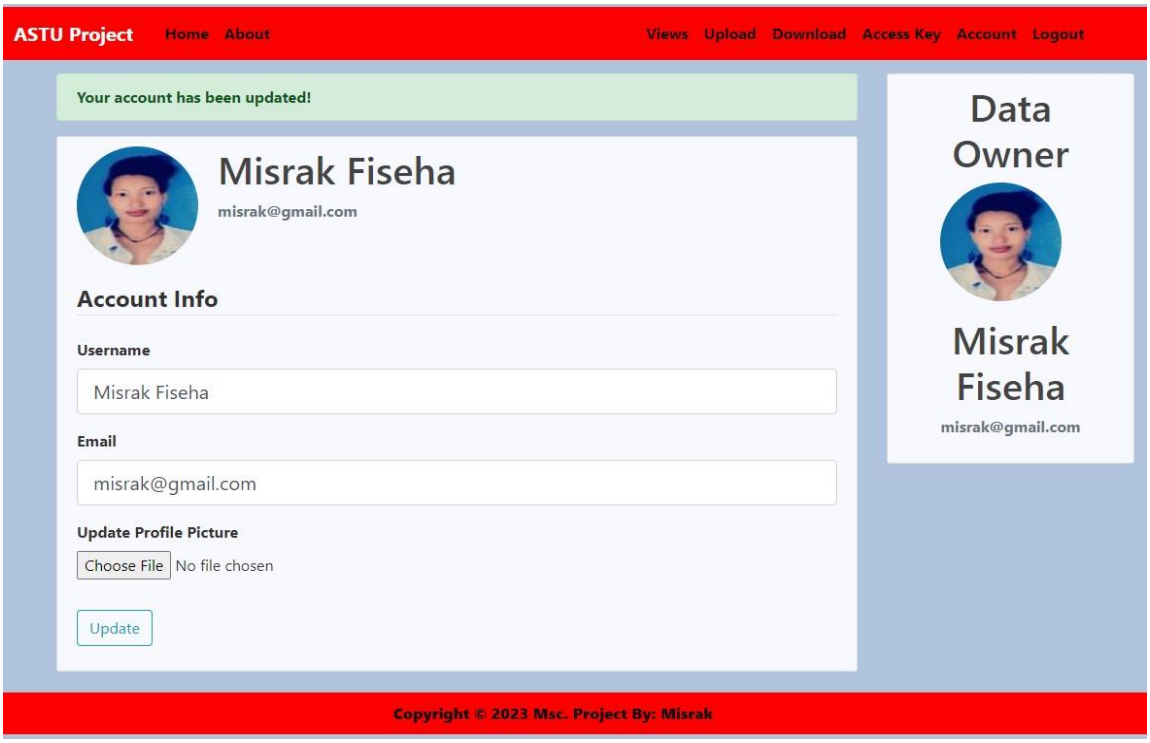


Figure 4. 3: User login form



Default Profile Picture

Figure 4. 4: User account operation page.

4.2.2. File Uploading Process

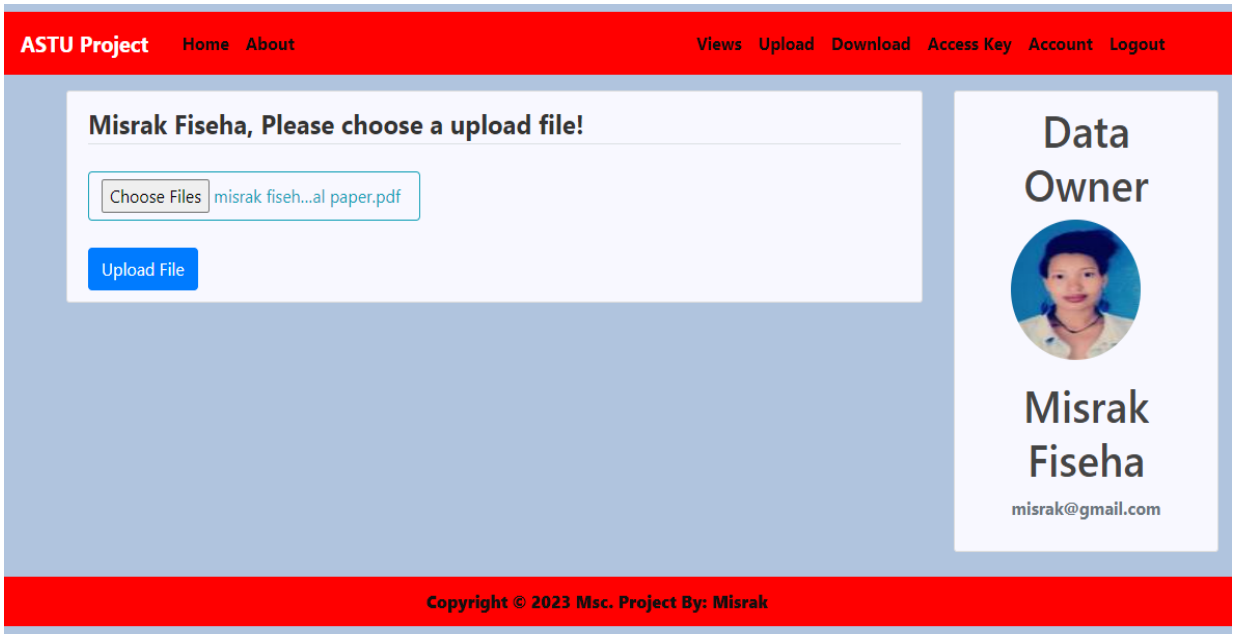
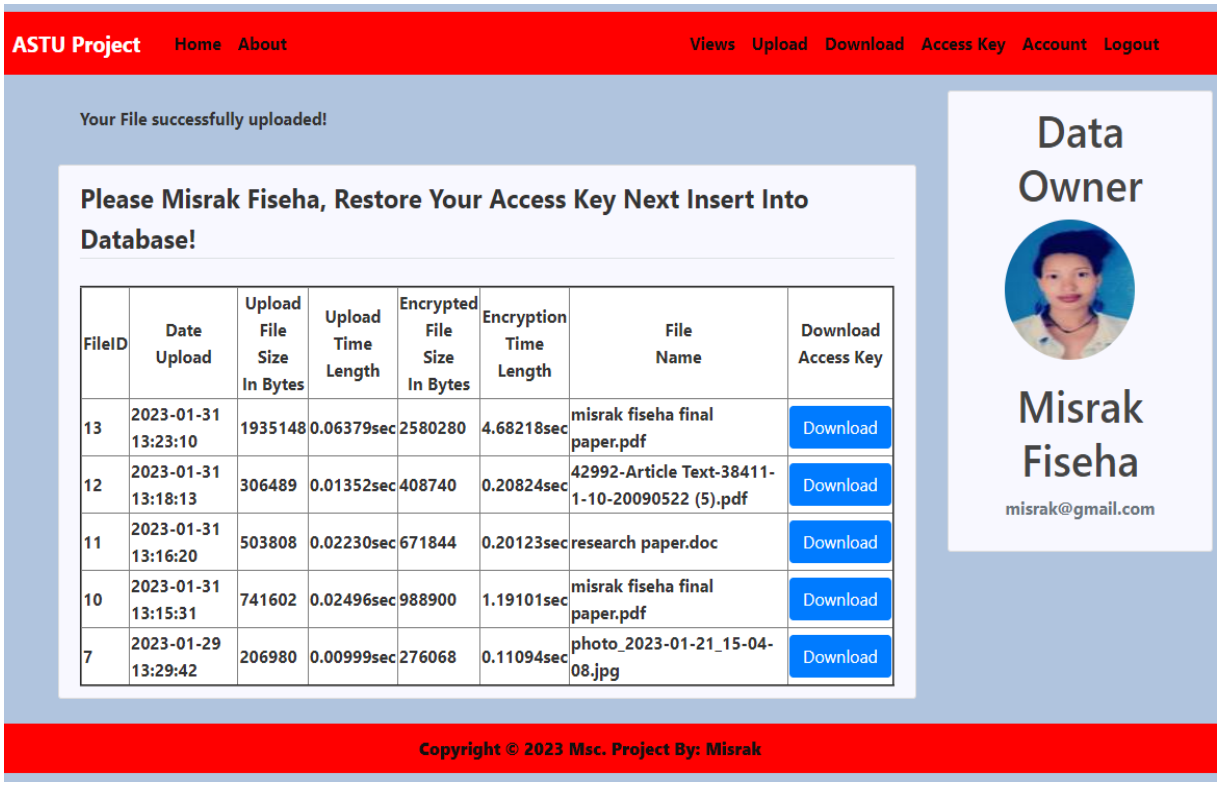


Figure 4. 5: Upload new file



FileID	Date Upload	Upload File Size In Bytes	Upload Time Length	Encrypted File Size In Bytes	Encryption Time Length	File Name	Download Access Key
13	2023-01-31 13:23:10	1935148	0.06379sec	2580280	4.68218sec	misrak fiseha final paper.pdf	Download
12	2023-01-31 13:18:13	306489	0.01352sec	408740	0.20824sec	42992-Article Text-38411-1-10-20090522 (5).pdf	Download
11	2023-01-31 13:16:20	503808	0.02230sec	671844	0.20123sec	research paper.doc	Download
10	2023-01-31 13:15:31	741602	0.02496sec	988900	1.19101sec	misrak fiseha final paper.pdf	Download
7	2023-01-29 13:29:42	206980	0.00999sec	276068	0.11094sec	photo_2023-01-21_15-04-08.jpg	Download

Figure 4. 6: Successfully upload file and restore access key

4.2.3. Insert or Save Access Key to Database

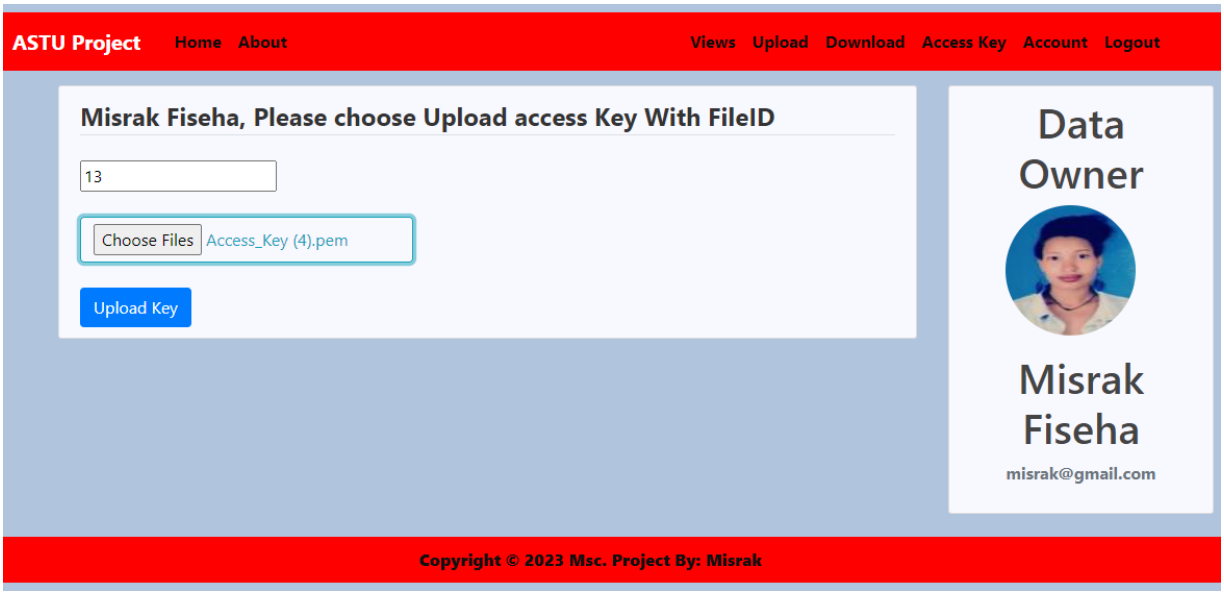


Figure 4. 7: Store access key with File ID to database

4.2.4. File Downloading Process

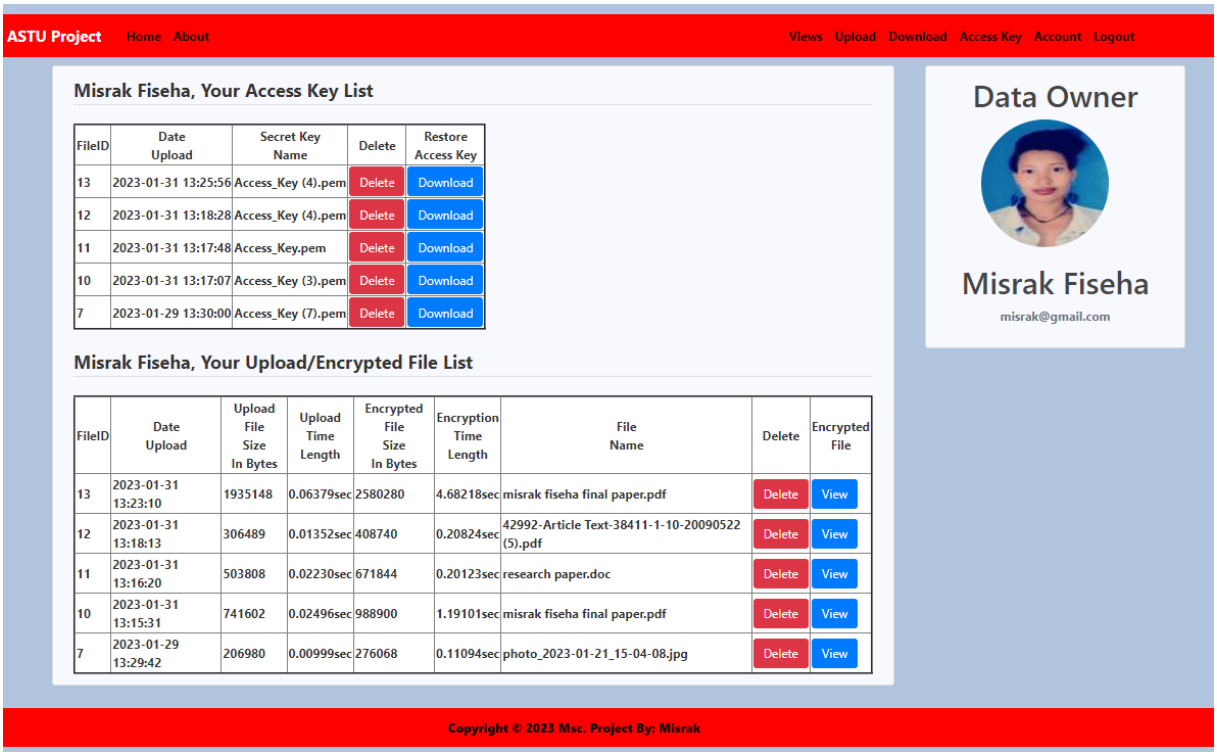


Figure 4. 8: Restore access key on selected file from view page

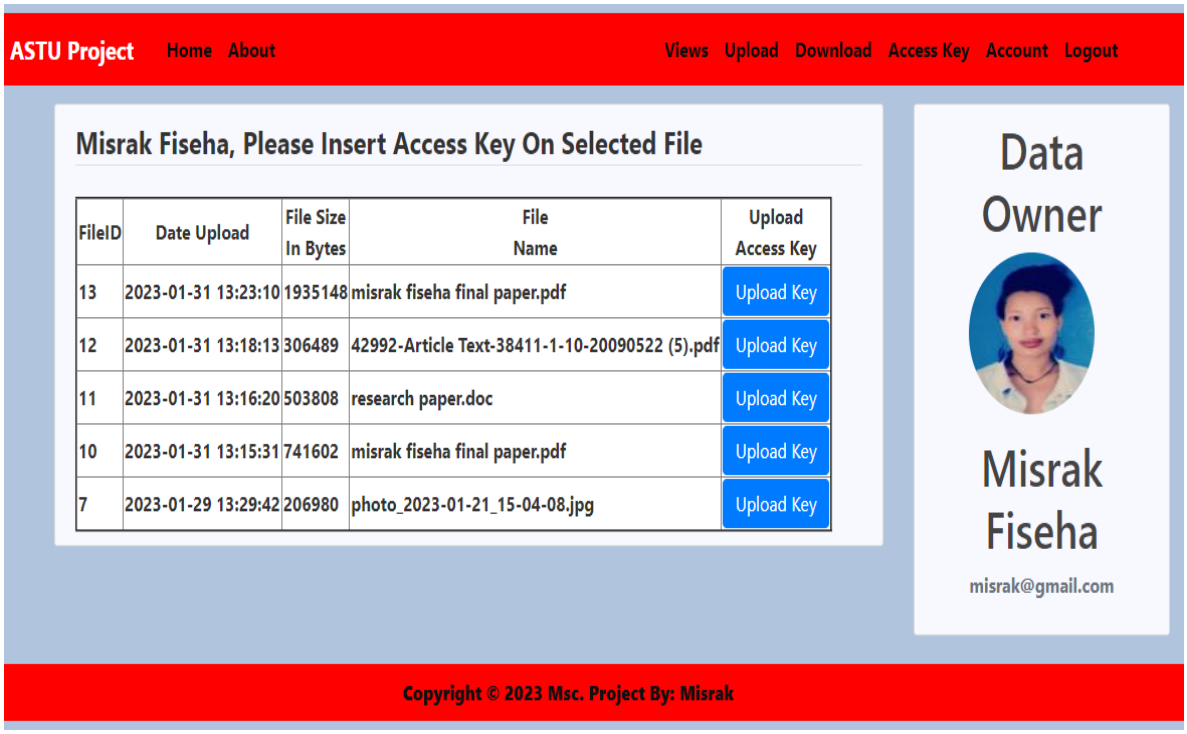


Figure 4. 9: Upload access key on selected file from download page

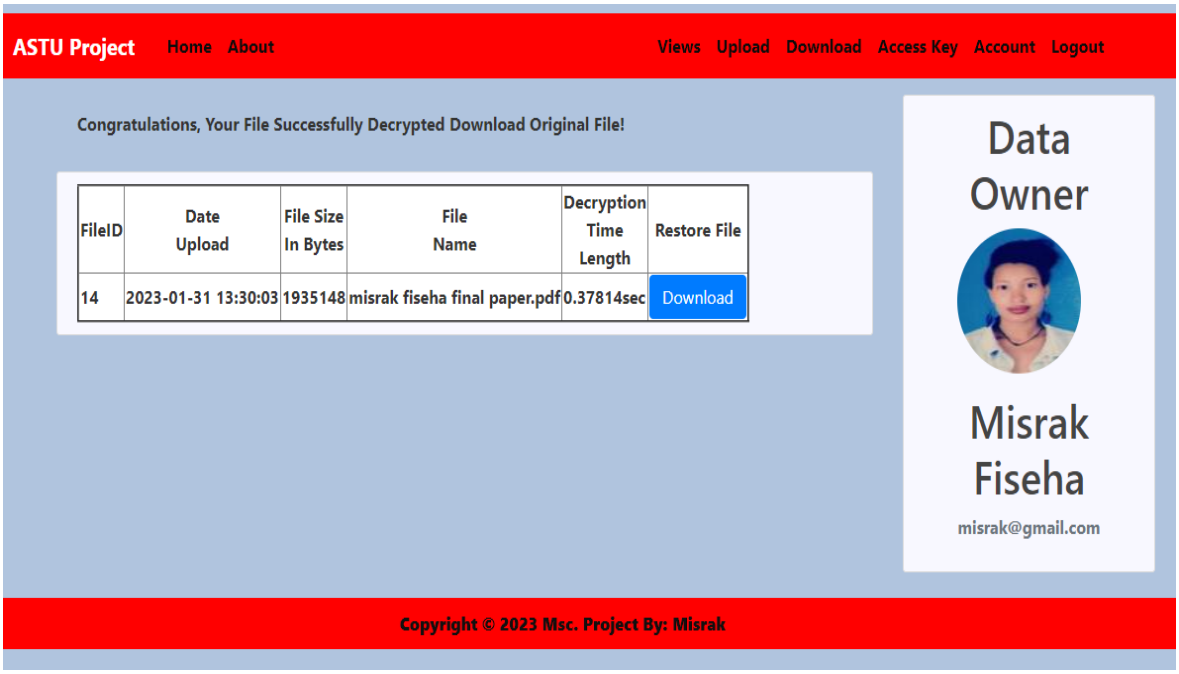


Figure 4. 10: Download original /decrypted file

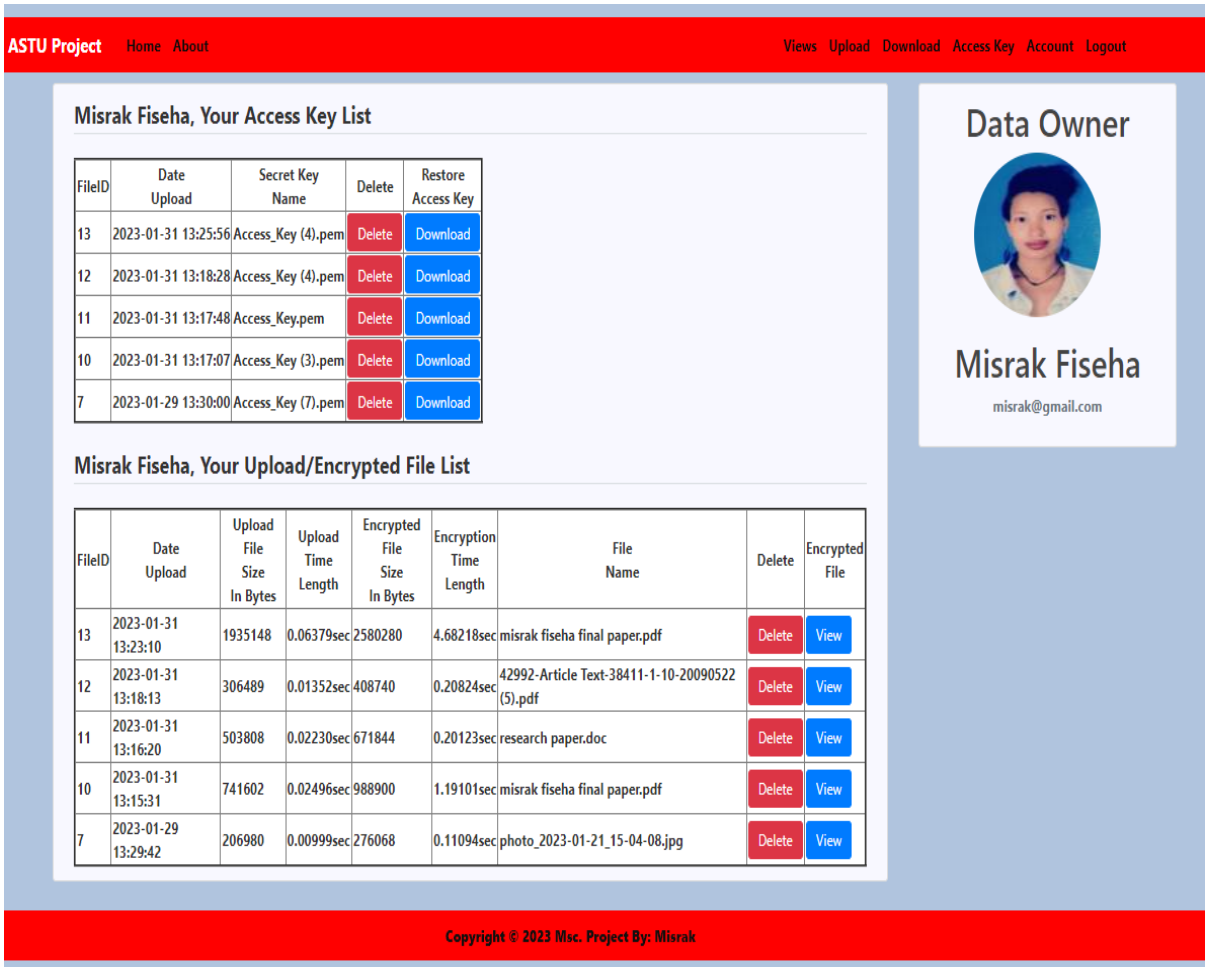


Figure 4. 11: Data owner view all access key and upload file list from view page

CHAPTER FIVE

EXPERIMENTAL RESULTS AND EVALUATION

5.1. Introduction

In the proposed work has been implemented as a web service available for use by every registered user after completing the sign up process and passing the login process using a web site designed and implemented for this purpose. In the proposed framework, the data owner can access the system through any web browser such as Google chrome, Firefox, or Internet Explorer, etc. The technologies used for building the proposed system includes: Python, HTML, CSS, Java Script, MySQL, and PyCharm as the used integrated development environment. The experiments are conducted to measure the upload time, key generation time, and encryption time by the help of Prometheus performance metrics.

5.2. Experiments

The following experiments consist of 120 different files size. The types, numbers, and sources of the files are shown in detail in table 5.1.

Table 5. 1: The details of the files size used to evaluate the performance of the proposed system

Sequence	File Type	Number of Fil	View Source
1	Document File	24	Microsoft Word
2	Audio	15	YouTube
3	Video	20	YouTube
4	Image File	20	Internet
5	PDF File	22	Internet
6	PowerPoint File	19	Microsoft PowerPoint

5.3. Results and Evaluation

In this section, a set of experiments have been executed to evaluate the performance of the proposed system. The first five experiments have been done on the data described in Table5.2, consists of 20 files with different sizes and different types. The first five experiments are conducted to measure the upload time, key generation time, encryption time, encrypted file size and decryption time sequentially.

Table 5. 2: The details of the file used in the first five experiments and the results of the different experiments.

No	File Type	File Size KB	Upload Time Milliseconds		Key Generation Time Milliseconds	Encryption Time Milliseconds	Encrypted File Size KB	Decryption Time Milli-seconds
			Public File	Private File				
1	PowerPoint	246.4	38.13	35.11	111.22	118.77	328.696	37.87
2	Pdf	290.8	44.43	42.34	132.37	136.52	387.956	47.83
3	Pdf	467.1	51.39	48.29	162.60	163.75	622.884	49.66
4	PowerPoint	497.7	56.28	54.65	196.09	198.99	663.745	55.98
5	Pdf	500.1	59.17	57.33	205.82	207.22	666.696	57.65
6	Pdf	510.5	60.87	58.91	212.75	213.17	680.804	58.28
7	PowerPoint	609.2	64.00	62.85	226.16	226.80	812.408	63.01
8	Word	685.4	70.68	68.44	242.62	243.38	913.934	70.20
9	Word	695.1	72.68	70.00	246.44	247.08	926.975	73.11
10	Word	866.1	75.23	73.55	344.30	344.84	1154.851	85.30
11	Word	954.2	76.27	74.97	461.01	471.93	1287.343	88.87
12	Word	1022	77.27	76.53	481.01	491.93	1362.725	97.89
13	Pdf	1100	88.99	86.63	621.36	626.47	1478.133	98.87
14	Pdf	1200	94.46	92.79	800.00	800.58	1576.096	99.89
15	Pdf	1300	120.71	115.43	812.46	814.11	1678.975	115.30
16	Pdf	1900	143.12	138.92	1790.28	1791.28	2485.674	122.99
17	Pdf	2000	167.76	164.54	2680.99	2681.99	2692.547	130.36
18	Word	2100	169.43	164.35	4650.94	4651.64	2798.346	146.20
19	Word	7200	203.43	198.45	6896.77	6898.67	9567.452	278.15
20	Video	7600	401.73	395.21	28447.06	28451.08	10033.895	297.71

As mentioned previously, the first experiment has been done to compute the time (in milliseconds) needed to upload a set of files which have different sizes (in KB). Upload time is the time required to upload a specific file to the cloud storage, i.e. it is the difference between the start upload time and finish upload time. In addition to Table 5.2, the results of the first experiment are shown in Figure 5.1.

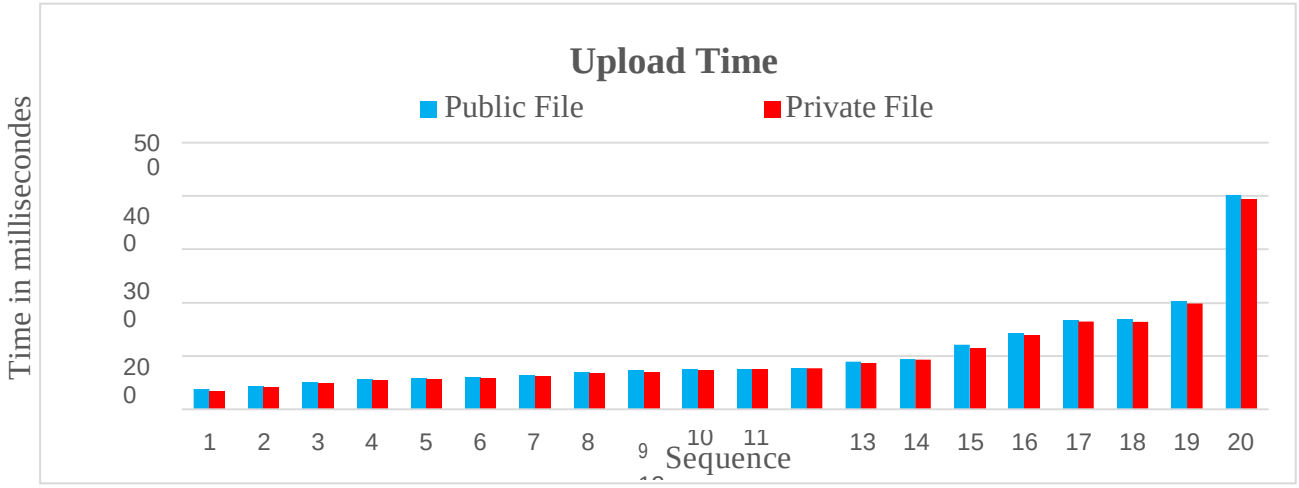


Figure 5. 1: The upload time for a set of files with different sizes.

The second experiment has been done to compute the time required to generate the encryption keys for each file to be encrypted using the AES encryption algorithm. In addition to Table 5.2, the results of the second experiments are shown in Figure 5.2. The third experiment computes the time of the encryption for each file in the set and the results are shown in Figure 5.3. Both of the experiments are applied on 20 files.

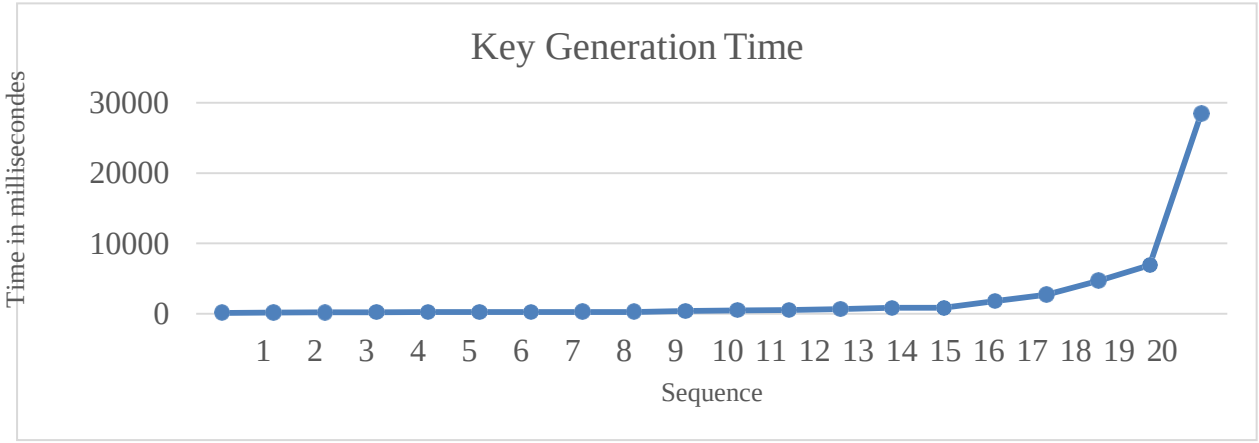


Figure 5. 2: The time of generating the encryption keys for a set of files with different sizes

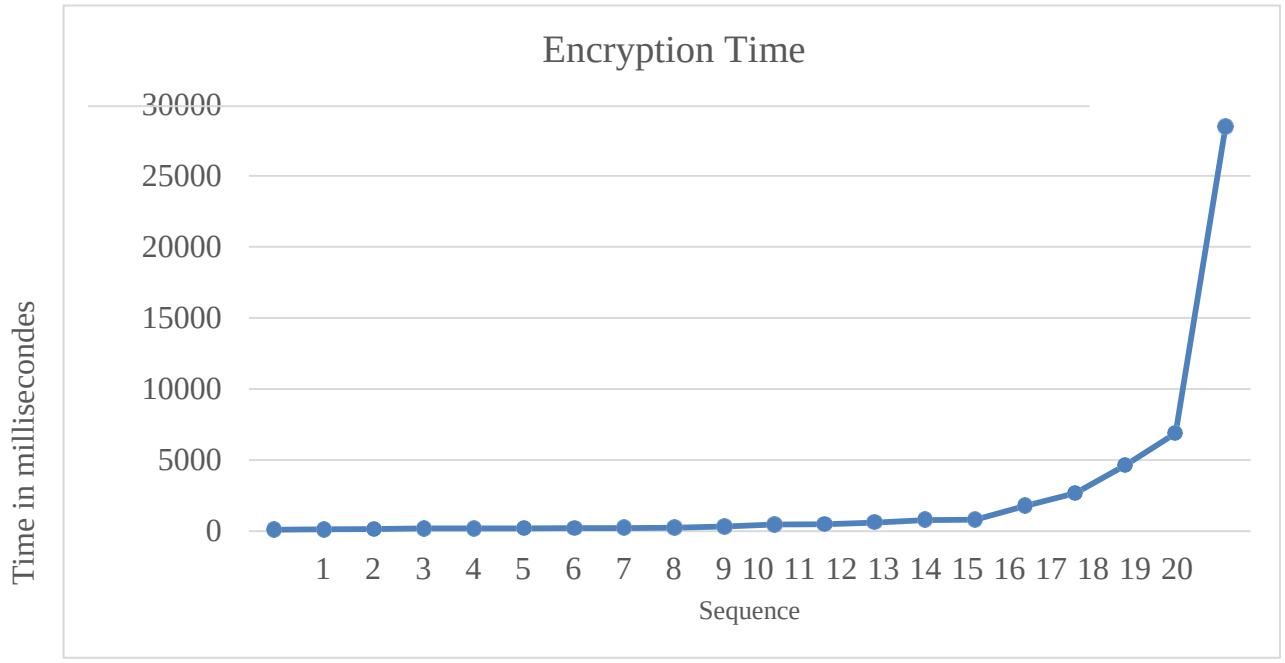


Figure 5. 3: The time of encryption using the AES for a set of files with different sizes.

From the previous Figures, it can be seen that the upload time, key generation time, and encryption time are directly proportional to the file size. The fourth experiment has been done the file size after encryption. The results in KB are shown in Figure 5.4.

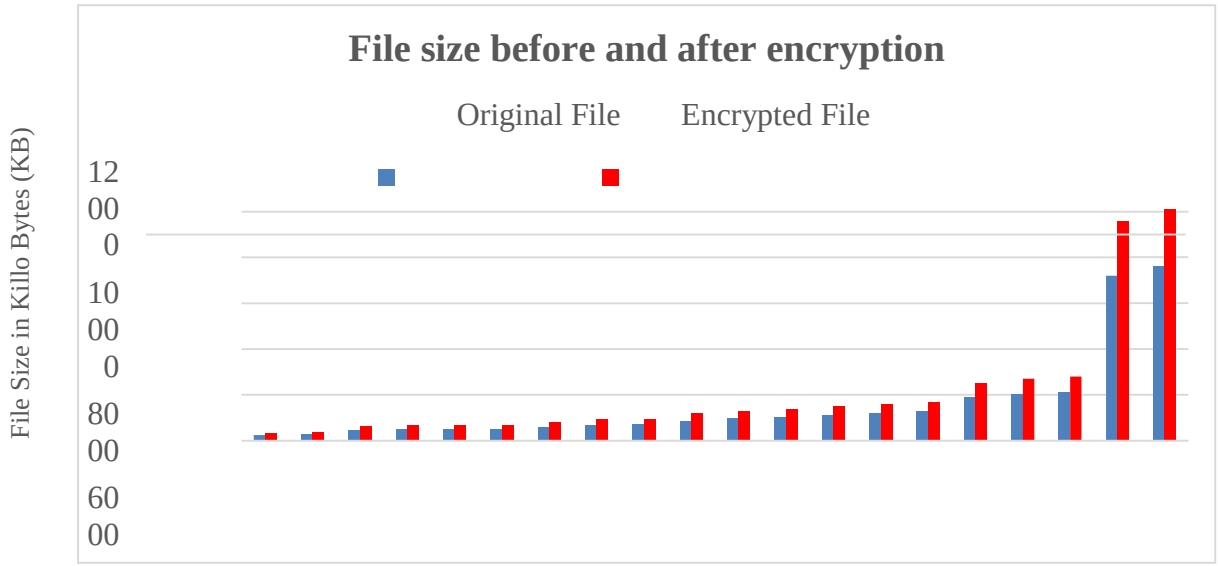


Figure 5. 4: The file size before and after encryption.

The fifth experiment has been done the decryption time. The results in milliseconds are shown in Figure 5.5.

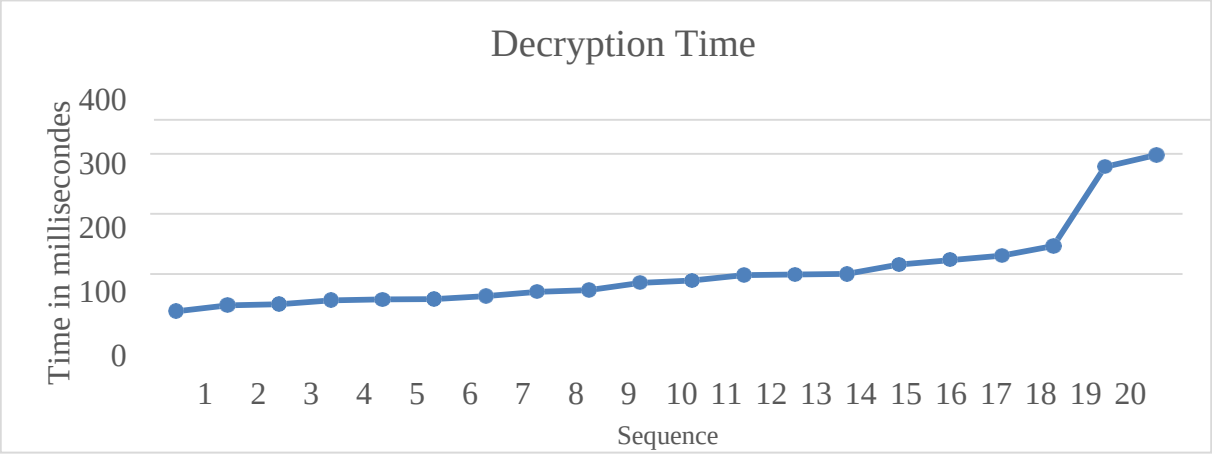


Figure 5. 5: The time of decryption using the AES for a set of files with different sizes.

5.4. Security Strength

Based on the encryption and decryption processes, the data security of the proposed system provides increase security strength based on the following factors:

- The data are encrypted by AES algorithm,
- The AES keys individually encrypted with the public key of RSA algorithm.
- For data verification using RSA Digital Signature algorithm.
- The result in a hybrid that are hard to break because of unbreakable AES key and 1024 bit of RSA algorithm.

With the above stated factors, we considered the proposed system are increased security and simplicity.

5.5. Comparisons

In this section we compare the performance of the proposed system and the existing system. The existence system uses AES algorithm to encrypt the file at the time when storing data in the database. Thus, provides less security, because to maintain the security of data/information in cloud storage system provides an encryption mechanism using only AES algorithm and also key management problem.

To evaluate the performance of each system, we computed the security strength, time

efficiency which is measured based on the upload time, and encryption time for different data files as shown table 5.3.

Table 5. 3: Size of the files used to measure time efficiency of the upload time and encryption time process.

Sequence	File Type	File Size KB	Upload Time		Encryption Time	
			Existing System Milliseconds	Proposed System Milliseconds	Existing System Milliseconds	Proposed System Milliseconds
1	Text	10	178	120	161	98
2	Text	14	189	125	170	105
3	Text	16	200	153	180	111
4	Text	22	203	155	191	119
5	Text	23	208	155	211	128
6	Text	31	210	158	213	131
7	Word	37	211	161	215	165
8	Word	125	219	173	217	169

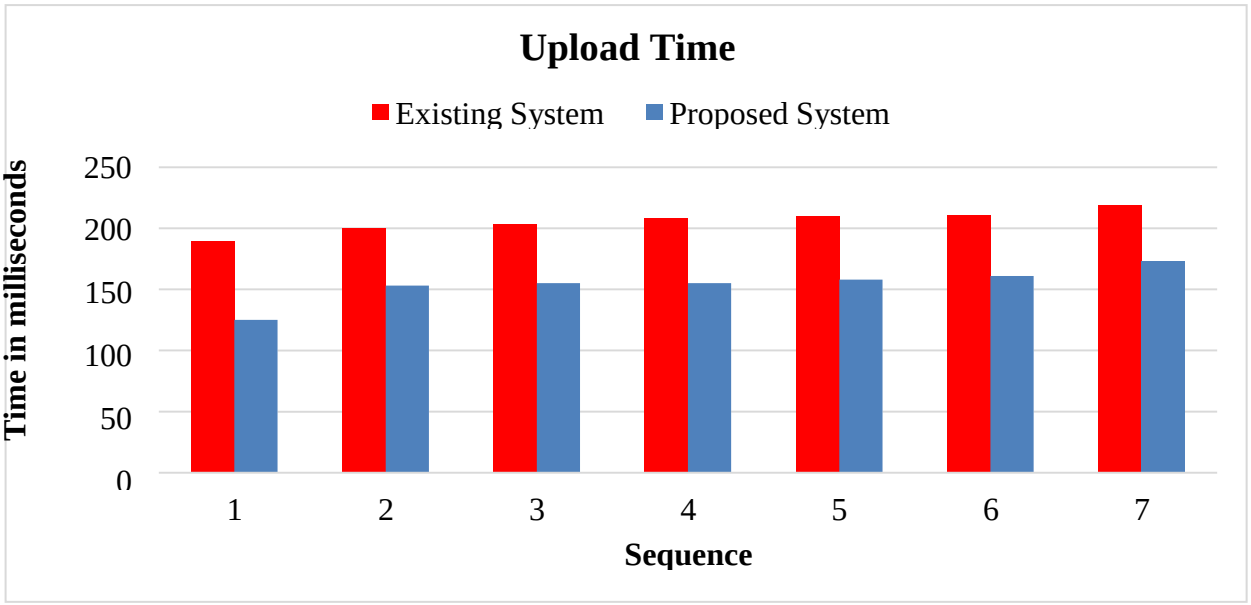


Figure 5. 6: The upload time of the existing system and proposed system

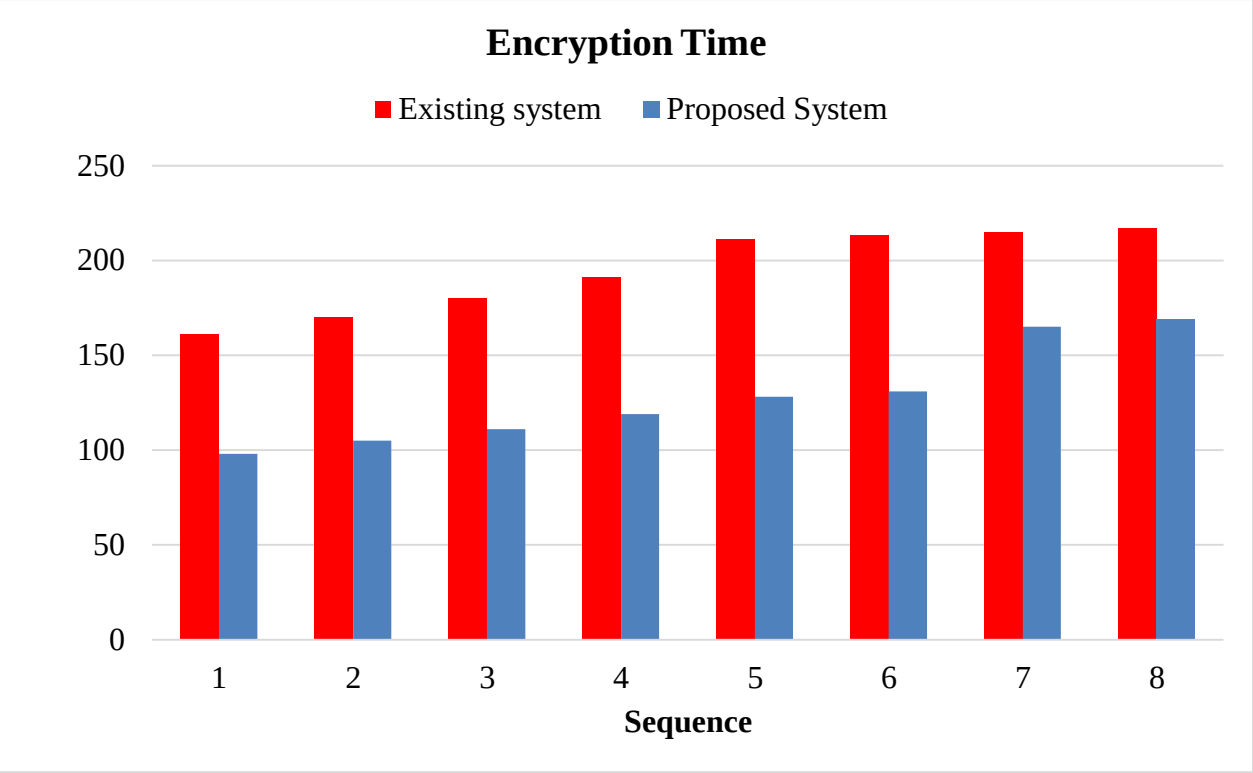


Figure 5. 7: The encryption time of the existing system and proposed system.

Figure 5.6 and Figure 5.6 shows that the proposed method is faster than the Existing System.

Therefore, the proposed method is more efficient in terms of the required time for performing the upload time and encryption time.

5.6. Summary

Moreover, the proposed schemes have a better time efficiency than the existing system while the data size was observed to slightly increase after encryption as shown Figure 5.4 but remained same as the original after decryption. Also, the proposed schemes were found to have better security strength. Consequently, the proposed schemes could be more preferable in ensuring the confidentiality, and integrity of cloud users' data which in turn will either reduce or eliminate existing uncertainty that delays well-known cloud adoption by individuals and organizations.

CHAPTER SIX

CONCLUSIONS AND FUTURE WORKS

6.1. Conclusion

In this paper we have proposed on focus on security and privacy of data on cloud computing using hybrid cryptography techniques. This thesis works is to optimize the security of the data storage on cloud using hybrid cryptosystem (AES & RSA) algorithm so as to have controlled authenticated and authorized access. In order to provide security for our encryption file we apply symmetric AES algorithm with key length of 128 bit which is one of the symmetric algorithms recommended by NIST and provide more security than other algorithms. RSA algorithm is used to encrypt AES key and digital signature this achieve authorization service.

In this work the details on the literatures reviewed related to this thesis is described. and brief comparative summary of literature reviewed indicating the research gap is explained. After that the proposed schemes is designed and the techniques used in the study was reviewed. And this technique was successfully implemented using Python programming language in Pycharm IDE environment, and also the proposed schemes deployed on Heroku platform.

Finally, implementation results and analysis of the experimental work is examined. Based on the implementation results and evaluations, we have measure the upload time, data encryption time and the file size after encryption with 120 different file size and plotted graph respectively. On plotting the graph, we studied that upload time, encryption time and file size after encryption are directly proportional to each other. Moreover, the proposed schemes have a better time efficiency than the existing system.

Generally, the proposed system is very important for all sectors like governments, non-profits or small businesses and companies that need high level security for keeping their secret data from any kinds of harmful threats during data storage on the private cloud, because of its increased security and simplicity.

6.2. Future Works

In future research it would be appropriate to continue the project expanding the potentialities of the framework, such as testing other compression algorithms, implementing new security methods, test the functioning with other users and make comparative tests and expand the scope of the framework implementing more cloud service.

REFERENCES

Vlad Bucur;Ovidiu Stan;Liviu C. Miclea2019 22nd International Conference on Control Systems and Computer Science (CSCS) Data Loss Prevention and Data Protection in Cloud Environments Based on Authentication Tokens Year: 2019 | Conference Paper | Publisher: IEEE https://thesai.org/Publications/ViewPaper?Volume=12& Issue=6& Code=IJACSA&SerialNo=4

Foundation, O. (2017a). Monasca, https://wiki.openstack.org/wiki/monasca.

Lu, C. C., & Tseng, S. Y. (2012). Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter. In Application-Specific Systems, Architectures and Processors,. Proceedings. The IEEE International Conference on (pp. 277-285)

Sattar J Aboudet al. (April 2008), "An Efficient RSA Public Key Encryption Scheme", Proc. of IEEE Fifth International Conference on Information Technology New Generations, pp127-130.

Jitendra Singh Yadavet, "Modified RSA Public Key Cryptosystem Using Short Range Natura Number Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, vol.2. Aug 2012

2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) Venkata Ramaiah Kavuri;Anithaashri T. P. Efficient Secured Cloud Storage System using Dynamic Multiple Clouds Cryptographic Algorithm Year: 2023 | Conference Paper | Publisher: IEEE

Chetan Gaikwad, Bhoomika Churi, Kanad Patiland Tatwadarshi P. N. "Providing Storage as a Service on Cloud using OpenStack", IEEE, 2017.

V.S. Mahalle, A. K. Shahade, "Enhancing the Data Security in Cloud by Implementing Hybrid (Rsa & Aes) Encryption Algorithm", IEEE, INPAC,pp 146-149,Oct .2014.

Jasleen K., S.Garg," *Security in Cloud Computing using Hybrid of Algorithms"* IEEE pages 300-305, September-October, 2015.

S. Yadav, "Comparative Study on *Open Source Software for Cloud Computing Platform: Eucalyptus, Openstack and Opennebula*", International Journal Of Engineering And Science, Vol.3, Issue 10, 2013, pp 51-54.

Punam V. Maitri, Aruna Verma," *Secure File storage in Cloud Computing using Hybrid*

Cryptography Algorithm", IEEE, International Conference on Reliability, Optimization and Information Technology, pages 1635-1638 ,2016.

Vlad Bucur;Ovidiu Stan;Liviu C. Miclea 2019 22nd International Conference on Control Systems and Computer Science (CSCS) *Data Loss Prevention and Data Protection in Cloud Environments Based on Authentication Tokens* Year: 2019 | Conference Paper | Publisher: IEEE

Kawser Wazed Naf, Tonny Shekha Kar, Sayed Anisul Hoque, and Dr. M. M. A Hashem,"*A Newer User Authentication, File encryption and Distributed Server Based Cloud Computing security architecture*",International Journal of Advanced Computer Science and Applications, Vol. 3, No. 10,2012

D.Mukhopadhyay,"*Enhanced Security for Cloud Storage using File Encryption",* Proc. International Conference on Distributed Computing and Internet Technologies,2013

Prateeksha Jain;Vinay Kumar Mishra;Vivek Kumar 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART) Year: 2022 | *An Enhanced Algorithm to Improve the Security in Cloud using Hybrid Hash Algorithm* Conference Paper | Publisher: IEEE

Mustafa S. Abbas;Suadad S. Mahdi;Shahad A. Hussien 2020 International Conference on Computer Science and Software Engineering (CSASE)Year: 2020 | *Security Improvement of Cloud Data Using Hybrid Cryptography and Steganography* Conference Paper | Publisher: IEEE

Bashir Mohammed and Mariam Kiran "Analysis *of Cloud Test Beds using Open Source Solutions ",* 3rd International Conference on Future Internet of Things and Cloud, 2015.

https://ieeexplore.ieee.org/document/8878629

Ertaul, L., A. Movasseghi and S. Kumar, "*Enterprise Security Planning with TOGAF- 72 9," Math & Computer Science,* CSU East Bay, Hayward, CA, USA

John Sherwood, Andrew Clark, and David Lynas, "Enterprise Security Architecture,"

SABSA, White paper, SABSA Limited, 1995-2009

J. Yadav et al., "Modified RSA Public Key Cryptosystem Using Short Range Natural

Number Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, August 2012.

Ako Muhammad Abdullah, "Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data", July 2017.

Yenugu vanilanka, J., & Elkeelany, O. (2008, April). Performance evaluation of hardware models of Advanced Encryption Standard (AES) algorithm. In Southeast con, 2008. IEEE (pp. 222-225).

Lu, C. C., & Tseng, S. Y. (2002). Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter. In Application-Specific Systems, Architectures and Processors, 2002. Proceedings. The IEEE International Conference on (pp. 277-285).

https://docs.python.org/3/license.html, September 26, 2020.

https://en.wikipedia.org/wiki/MySQL, last access on: September 26, 2019.

https://www.openstack.org/software, last access on: September 27, 2019. *Secure algorithm for cloud computing and its applications*

Published in: 2022 11th International Conference on System Modeling & Advancement in Research Trends. Published in: 2022

https://ieeexplore.ieee.org/document/8878629https://thesai.org/Publications/ViewPaper?Volume=12&Issue=6&Code=IJACSA&SerialNo=4https://ieeexplore.ieee.org/document/10072437

Appendix

A. SAMPLE PYTHON CODES #

```
# __init__.y
import os
 import uuid
import shutil from flask
import Flaskfrom flask_sqlalchemy
 import SQLAlchemy
 from flask_bcrypt  import Bcrypt
from flask_login
import LoginManager from flask_mail
import Mail from flask_script
 import Manager from flask_migrate
 import MigrateCommand, Migrate
UPLOAD_FOLDER = '. /uploads/'
UPLOAD_KEY = '. /mainproject/key/'
#UPLOAD_FOLDER = '. /uploads/'
#UPLOAD_KEY = '. /key/
RSTORED_FOLDER = '. /restored_file/'
ALLOWED_EXTENSIONS = {'txt', 'doc', 'docx', 'ppt', 'pptx', 'pdf', 'pgp', 'gif', 'jpg', 'mp3',
'mp4'} ALLOW_EXTENSIONS = {'pem'}
app = Flask(__name__)
SECRET_KEY = os.urandom (24) app.config
['SECRET_KEY'] = SECRET_KEY
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql+pymysql:
//cloud_storage:123den@localhost/hybrid_db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = True
```