

Project Report

On

Symmetric-Key Encryption for Cloud Storage Security

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Engineering

In

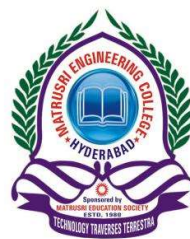
Computer Science and Engineering

By

B Aishwarya	160820733069
S Bhavya Sri	160820733081
V Sai Prakash	160820733080

Under the guidance of

Mrs B Sonal
Assistant Professor



Department of Computer Science and Engineering

Matrusri Engineering College

Accredited by NBA & NAAC

(Affiliated to Osmania University, Approved by AICTE)

Saidabad, Hyderabad-500059

2023-2024

Department of Computer Science and Engineering

Matrusri Engineering College

Accredited by NBA & NAAC

(Affiliated to Osmania University, Approved by AICTE)

Saidabad, Hyderabad-500059

2023-2024



CERTIFICATE

This is to Certify that a Project report entitled “**Symmetric-Key Encryption for Cloud Storage Security**” is being submitted by B Aishwarya (1608-20-733-069), S Bhavya Sri (1608-20-733-081), and V Sai Prakash (1608-20-733-080) in partial fulfillment of the requirement of the award for the degree of Bachelor of Engineering in “Computer Science and Engineering” O.U., Hyderabad during the year 2023-2024 is a record of bonafide work carried out by him/her under my guidance. The results presented in this thesis have been verified and are found to be satisfactory.

Project Guide

Mrs. B.Sonal

Assistant Professor,

Dept of C.S.E.

H.O.D.

Dr. P. Vijayapal Reddy

Professor & Head,

Dept. of CSE

External Examiner(s)

MATRUSRI ENGINEERING COLLEGE
SAIDABAD – 500059



Department of Computer Science and Engineering

DECLARATION BY THE CANDIDATE

We, Badampet Aishwarya (1608-20-733-069), Sattu Bhavya Sri (1608-20-733-081), Vaddeboina Sai Prakash (1608-20-733-080), hereby certify that the major project report entitled “Symmetric-Key Encryption for Cloud Storage Security” is submitted in the partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering in Computer Science and Engineering.

This is a Record of bonafide work carried out by us under the guidance of Mrs. B.Sonal , Assistant Professor, Matrusri Engineering College, Saidabad. The results embodied in this report have not been reproduced/copied from any source. The results embodied in this report have not been submitted to any other university or institute for the award of any other degree or diploma.

Badampet Aishwarya (1608-20-733-069)

Sattu Bhavya Sri (1608-20-733-081)

Vaddeboina Sai Prakash (1608-20-733-080)

ACKNOWLEDGEMENT

We wish to take this opportunity to express our deep gratitude to all the people who have extended their cooperation in various ways during our major project work. It's our pleasure to acknowledge the help of all those individuals.

Firstly, We would like to thank **Dr. P. Vijaya Pal Reddy (HOD, DEPT OF CSE.)** for his encouragement and valuable guidance in bringing shape to dissertation.

We would like to thank my project guide, Mrs. B. Sonal and for her guidance and help throughout the development of this project work by providing me with required information and support. Without her guidance, cooperation, and encouragement, I couldn't learn many new things during my major project tenure.

B Aishwarya (1608-20-733-069)

S Bhavya Sri (1608-20-733-081)

V Sai Prakash (1608-20-733-080)

ABSTRACT

Cloud encryption is the process of transforming data from its original plain text format to an unreadable format, such as ciphertext, before it is transferred to and stored in the cloud. As with any form of data encryption, cloud encryption renders the information indecipherable and therefore useless without the encryption keys. This applies even if the data is lost, stolen or shared with an unauthorized user. Encryption is regarded as one of the most effective components within the organization's cybersecurity strategy.

Cloud storage security is a collection of security measures designed to protect cloud-based infrastructure, applications, and data. One of the most critical aspects of cloud security is to ensure that only authorized personnel have access to the documents and files stored in the cloud. Symmetric encryption is also called "secret key" encryption because the key must be kept secret from third parties. Strengths of this method include speed and cryptographic strength per bit of key. Compared to asymmetric encryption, symmetric encryption is efficient as it is used for handling large amounts of data and requires more amount of time for cracking.

Cloud environments are vulnerable to network-based attacks, as tactics like distributed denial-of-service (DDoS) can throttle speed, block traffic, and cause unpredictable downtime. Unauthorized access can take longer to detect on the cloud than on physical, on-premise systems. This paper proposes an efficient mechanism for cloud storages using symmetric key encryption algorithm called Advanced encryption standard(AES) with the help of secret key.

CONTENTS

Declaration	iii	
Acknowledgement	iv	
Abstract	v	
List of Figures	viii	
S.No	Chapter	Page No
1.	INTRODUCTION	
	1.1 Overview	1
	1.2 Cloud Security and its Challenges	4
	1.3 Objectives	5
2.	LITERATURE SURVEY	7
3.	ANALYSIS	
	3.1 Problem Statement	14
	3.2 Existing System	14
	3.3 Symmetric-Key Encryption	15
	3.4 Proposed System	16
4.	DESIGN	
	4.1 UML Design	19
	4.1.1 Class Diagram	20
	4.1.2 Activity Diagram	21
	4.1.3 Use case Diagram	22
	4.1.4 Sequence Diagram	23
	4.1.5 Communication Diagram	24
	4.1.6 Component Diagram	25
	4.1.7 Deployment Diagram	26
	4.2 System Architecture	27

5.	IMPLEMENTATION	
	5.1 Java	29
	5.2 JSP	30
	5.3 CSS	31
	5.4 AES Algorithm	33
	5.5 Implementation of AES Algorithm	38
6.	TESTING	
	6.1 Testing the frontend and backend	41
	6.2 Testing the working of Secret Key	43
7.	RESULTS	45
8.	CONCLUSION	
	8.1 Conclusion	51
	8.2 Future Scope	52
	8.3 Reference	53

LIST OF FIGURES

S.No	Fig No.	Name of the Figure	Page No
1	3.1	Symmetric-Key Encryption	15
2	4.1	Class Diagram	20
3	4.2	Activity Diagram	21
4	4.3	Use Case Diagram	22
5	4.4	Sequence Diagram	23
6	4.5	Communication Diagram	24
7	4.6	Component Diagram	25
8	4.7	Deployment Diagram	26
9	4.8	System Architecture	27
10	5.1	AES	34
11	5.2	AES	34
12	5.3	Add Round Key	35
13	5.4	SubBytes	35
14	5.5	Shift Rows	36
15	5.6	Implementation of AES Algorithm	38
16	5.7	Implementation of AES Algorithm	39
17	6.1	Testing	41
18	6.2	Testing the Working of Secret Key	43
19	7.1	Main Page	45
20	7.2	Admin Login Page	45
21	7.3	Registration Form	46
22	7.4	Data Owner Registration Form	46
23	7.5	Data Owner Login Form	47
24	7.6	File Upload Page	47
25	7.7	User Registration Page	48
26	7.8	User Login Page	48
27	7.9	File Download Page	49

CHAPTER 1

1.INTRODUCTION

1.1 Overview

Security is defined as the state of being free from danger or threat, i.e the system is designed to provide maximum security against toxic spills. Cloud security is a collection of procedures and technology designed to address external and internal threats to business security. Organizations need cloud security as they move toward their digital transformation strategy and incorporate cloud-based tools and services as part of their infrastructure. The terms digital transformation and cloud migration have been used regularly in enterprise settings over recent years. While both phrases can mean different things to different organizations, each is driven by a common denominator: the need for change. As enterprises embrace these concepts and move toward optimizing their operational approach, new challenges arise when balancing productivity levels and security. While more modern technologies help organizations advance capabilities outside the confines of on-premises infrastructure, transitioning primarily to cloud-based environments can have several implications if not done securely. Striking the right balance requires an understanding of how modern-day enterprises can benefit from the use of interconnected cloud technologies while deploying the best cloud security practices.

The "cloud" or, more specifically, "cloud computing" refers to the process of accessing resources, software and databases over the internet and outside the confines of local hardware restrictions. This technology gives organizations flexibility when scaling their operations by offloading a portion, or majority, of their infrastructure management to third-party hosting providers.

The way to approach cloud security is different for every organization and can depend on several variables. However, the National Institute of Standards and Technology (NIST) has made a list of best practices that can be followed to establish a secure and sustainable cloud computing framework. The NIST has created necessary steps for every organization to self-assess their security preparedness and apply adequate preventative and recovery security measures to their systems. These principles are built on the NIST's five pillars of a cybersecurity framework: Identify, protect, detect, respond and recover.

Encryption is the method by which information is converted into secret code that hides the information's true meaning. Cloud encryption is a data security process in which plaintext data is encoded into unreadable ciphertext to help keep it secure in or between cloud environments. It is one of the most effective ways to uphold data privacy as well as protect cloud data in transit or at rest against cyberattacks. Cloud encryption protects sensitive information as it traverses the internet or rests in the cloud. Encryption algorithms can transform data of any type into an encoded format that requires a decryption key to decipher. This way, even if an attacker intercepts or exfiltrates the data, it's useless to them unless they can decrypt it. The two main algorithms used for encryption are: Symmetric key encryption and Asymmetric key encryption. To provide an authenticated security to cloud storage, we can make use of various encryption techniques such as symmetric key cryptography or public-key encryption. The main objective is to provide secure encryption for the documents using symmetric key encryption methodology.

Symmetric key encryption is a type of encryption where only one key (a secret key) is used to both encrypt and decrypt electronic data. The entities communicating via symmetric encryption must exchange the key so that it can be used in the decryption process. This encryption method differs from asymmetric encryption where a pair of keys - one public and one private - is used to encrypt and decrypt messages. By using symmetric encryption algorithms, data is "scrambled" so that it can't be understood by anyone who does not possess the secret key to decrypt it. Once the intended recipient who possesses the key has the message, the algorithm reverses its action so that the message is returned to its original readable form. Some examples of symmetric encryption algorithms include:

- AES (Advanced Encryption Standard)
- DES (Data Encryption Standard)
- IDEA (International Data Encryption Algorithm)

While symmetric encryption is an older method of encryption, it is faster and more efficient than asymmetric encryption, which takes a toll on networks due to performance issues with data size and heavy CPU use. Due to the better performance and faster speed of symmetric encryption (compared to asymmetric), symmetric cryptography is typically used for bulk encryption / encrypting large amounts of data, e.g. for database encryption. In the

case of a database, the secret key might only be available to the database itself to encrypt or decrypt. Industry-standard symmetric encryption is also less vulnerable to advances in quantum computing compared to the current standards for asymmetric algorithms (at the time of writing).

The most commonly used symmetric algorithm is the Advanced Encryption Standard (AES), which was originally known as Rijndael. This is the standard set by the U.S. National Institute of Standards and Technology in 2001 for the encryption of electronic data announced in U.S. FIPS PUB 197. This standard supersedes DES, which had been in use since 1977. Under NIST, the AES cipher has a block size of 128 bits, but can have three different key lengths as shown with AES-128, AES-192 and AES-256. The main usage purpose of the Advanced Encryption Standard (AES) is to encrypt data and protect it from unauthorized access. This is accomplished through the use of a cryptographic process key of various lengths. Depending on the length, this is labeled AES-128, AES-192, or AES-256.

In 1970, The National Bureau of Standards (NBS) in the United States needed a secret algorithm to encrypt critical government data. On this need, a new symmetric key algorithm: Data Encryption Standard (DES) has been revealed. In 1997, The DES algorithm was discovered to be vulnerable to brute-force attacks. The National Institute of Standards and Technology (NIST) has launched a public competition to discover an alternative for the DES algorithm. In 1998, 15 different groups from all over the world participated in the contest organized by NIST by submitting their own AES algorithms. In 2001, The Rijndael group was chosen by NIST as the winner of the AES competition. After all long-term selection and validation tests, The Advanced Encryption Standard (AES) was published as FIPS 197 on November 26, 2001. The fact that Rijndael is royalty-free and that it can be easily implemented on a wide range of systems without significantly lowering bandwidth are two main drivers in its rapid adoption.

1.2 Cloud security and its challenges

Although it can greatly ease security management and increase visibility, cloud security comes with its share of challenges, underscoring how important it is to find the right partner.

1. Lack of cloud security strategy and skills

Traditional data center security models are not suitable for the cloud. Administrators must learn new strategies and skills specific to cloud computing.

Cloud may give organizations agility, but it can also open up vulnerabilities for organizations that lack the internal knowledge and skills to understand security challenges in the cloud effectively. Poor planning can manifest itself in misunderstanding the implications of the shared responsibility model, which lays out the security duties of the cloud provider and the user. This misunderstanding could lead to the exploitation of unintentional security holes.

2. Identity and access management

Identity and access management is essential. While this may seem obvious, the challenge lies in the details. It's a daunting task to create the necessary roles and permissions for an enterprise of thousands of employees. There are three steps to a holistic IAM strategy: role design, privileged access management, and implementation.

3.cloud ransomware

Cloud environments are still vulnerable to cyberattacks. Attackers most commonly infiltrate environments by taking advantage of misconfigurations or poor security practices, such as over-permissioned access, insufficient policy controls, or weak passwords.

4.logging, monitoring and incident response

Comprehensive and accurate logs are a cornerstone of effective incident response. Many organizations' existing solutions are ill-equipped for the volume of data cloud computing tends to produce, and are unable to reliably collect complete logs.

1.3 Objectives:

1. Confidentiality: This refers to the protection of data from unauthorized disclosure or access. Encryption algorithms can be used to encrypt data so that only authorized parties can access it.
 2. Integrity: This refers to the protection of data from unauthorized modification. Cryptography can be used to ensure that data has not been tampered with or altered in transit.
 3. Authentication: This refers to the process of verifying the identity of a user or system. Cryptography can be used to provide secure authentication mechanisms that prevent unauthorized access.
 4. Availability: This refers to the ability of a system to provide access to authorized users when they need it.
- ❑ To protect the data that is stored or moving in and out of the cloud from security threats, unauthorized access, theft, and corruption.
 - ❑ The main aim is to use symmetric key encryption algorithm called AES algorithm with the help of a secret key to ensure security of the data.

The main objective is to protect the data from Brute-force attacks, Snooping, Modification, Replaying, phishing, Dictionary attack and unauthorized user access or use. A brute-force attack is a type of password attack where hackers make numerous hit-or-miss attempts to gain access. It is a simple attack and often involves automated methods, such as software, for trying multiple letter-number variations. A replay attack (also known as a repeat attack or playback attack) is a form of an attack in which valid data transmission is maliciously or fraudulently repeated or delayed.

CHAPTER 2

2. LITERATURE SURVEY

[1]. Enhanced Public Key Encryption with Keyword Search in Cloud, Vol. 8 Issue International Journal of Engineering Research & Technology (IJERT) ISSN: 2278- 0181, Vol. 8 Issue 08, August-2019.

Cloud Computing is intended and supported primarily as a data center and an effective communication with the outside world. The main aim of the proposed framework is improving the LSPE model as far as expense and essentialness for creating the ciphertext. In the proposed structure, where the data sender scrambles a watchword as well as validates it to persuade a verifier that the encoded catchphrase must be created by the sender. In particular, we think about novel open key cryptography – Attribute-Based Encryption (ABE), and improve it toward giving an unquestionable cryptographic explanation behind a protected data sharing arrangement on untrusted amassing. In view of ABE, we likewise present our answers for verifying information partaking in Cloud Computing and remote sensor arrangements individually. Attribute based encryption (ABE) offers the capacity to scramble information without accurate learning of the collector set. In this sense the idea of ABE is firmly identified with RoleBased/Attribute-Based Access Control and reasonable for huge scale applications. Existing developments of ABE center around giving the fundamental functionalities, for example, information encryption/unscrambling and plot opposition. In Key Aggregate Cryptosystem, customers encode their information under an accessible door, yet what's more underneath an identifier of figure substance called class and those figure plays are additionally isolated into specific classes. The information proprietor holds a key called Master mystery key. The ace mystery can be used to create mystery keys for particular classes. All the more fundamentally, the produced key can be a total key which is as strong as a mystery key for a solitary class, yet consolidates the expert of numerous such keys, with the end goal that the decoding level for any subset of figure content classes. Sharing of these keys needs a secured channel and securing these keys need a sheltered storing. The cost and issues incorporate all things considered augmentations with a number of unscrambling keys to be shared. This project provides a Key Aggregate Cryptosystem along with the leakage resilience to data. The performance is promising and the verification of leaked data is efficiently handled by the key aggregate server.

[2].Double Server Public-Key Encryption with Keyword Search for Secure Cloud Storage,Volume 4 Issue 6, September-October 2020, e-ISSN: 2456 – 6470.

Cloud computing means storing and accessing the data and programs on remote servers that are hosted on the internet instead of the computer's hard drive or local server. Cloud computing is also referred to as Internet-based computing, it is a technology where the resource is provided as a service through the Internet to the user. The data which is stored can be files, images, documents, or any other storable document. In this task, explore the security of a notable cryptographic crude, specifically, public key encryption with watchword search (PEKS) which is valuable in numerous utilizations of distributed storage. As another fundamental commitment, characterize another variation of the smooth projective hash capacities alluded to as straight and homomorphic SPHF at that point show a conventional development of secure DS-PEKS from SPHF. The contributions of this project are four-fold. formalize a new PEKS framework named Dual-Server Public Key Encryption with Keyword Search (DS-PEKS) to address the security vulnerability of PEKS. A new variant of Smooth Projective Hash Function (SPHF), referred to as linear and homomorphic SPHF, is introduced for a generic construction of DS-PEKS. show a generic construction of DS-PEKS using the proposed Lin-Hom SPHF. To represent the plausibility of our new system, it gives an effective launch of the overall structure from a Decision Diffie–Hellman-based LH-SPHF and shows that it can accomplish the solid protection from inside the KGA. To illustrate the feasibility of our new framework, an efficient instantiation of our SPHF based on the Diffie- Hellman language is presented in this paper. All the existing schemes require the pairing computation during the generation of PEKS ciphertext and testing and hence are less efficient than our scheme, which does not need any pairing computation. Our scheme is the most efficient in terms of PEKS computation. In our paper it requires another stage for the testing, our computation cost is actually lower than that of any existing scheme as we do not require any pairing computation and all the searching work is handled by the server. In this paper, we proposed a new framework, named DualServer Public Key Encryption with Keyword Search (DSPEKS), that can prevent the inside keyword guessing attack which is an inherent vulnerability of the traditional PEKS framework. We also introduced a new Smooth Projective Hash Function (SPHF) and used it to construct a generic DSPEKS scheme. An efficient instantiation of the new SPHF based

on the Diffie-Hellman problem is also identified and solved. In the project, which gives secure data from an efficient DS-PEKS scheme with dynamic pairings.

[3].In-Depth Review on Keyword Search and Security Concepts,Volume 5, Issue 9, May-2018.

Keyword search techniques are outstandingly useful for separating both the structured and likewise the unstructured data which contains the broad measure of the literary information. In our research paper we will explore distinctive keyword search frameworks and we will in like manner endeavor to separate the domains on which we can work to upgrade execution of keyword search algorithms. By using Keyword Search customer can submit keyword to search engines (Internet Search) or structured data and subsequently it reestablishes an once-over of records to customer according to situating. Schema based methodologies bolster keyword search over social database (like SQL) using execution of SQL summons. These strategies are mix of vertices and edges including tuples and keys. By virtue of RDBMS, keyword search using the Schema Based Approach is performed through making use SQL. Data Spot is a database passing on instrument and it lets the end client to research the considerable database without making use of any demand vernacular. Token based techniques, such as key cards, bank cards and keen cards are generally utilized. Numerous token-based authentication frameworks likewise utilize knowledge based techniques to upgrade security. For instance, ATM cards are for the most part utilized together with a PIN number. The recognition-based framework considered most widely to date is Passfaces. For the most part amid setting a watchword the client chooses an arrangement of human appearances. A board of candidate faces is displayed amid his/her login. Among the given arrangement of distractions the client must choose the faces he/she chose amid setting the watchword. The photo based techniques can be additionally divided into two classes: recognition - based and review based graphical techniques. Utilizing recognition-based techniques, a client is given an arrangement of pictures and the client passes the authentication by perceiving and distinguishing the pictures he or she chose amid the enrollment organize. Utilizing review based techniques, a client is requested to duplicate something that he or she made or chose before amid the enlistment arrange. It is a tick based plan where clients select a single tick point on every one of 5 pictures in grouping, each one in turn; this gives one-to - one prompting. Amid the following login the client must recall that specific snap point on the offered picture to open the following right picture, if the snap

isn't right the following opened picture will be a phony one and not from the picked arrangement of pictures. This will stop current client authentication. This paper reviews the concepts of the Keyword Search and provides the slight reviews regarding the security models.

[4] DUAL - SERVER PUBLIC-KEY AUTHENTICATED ENCRYPTION WITH KEYWORD SEARCH, Mr. M. NAGARASAN, Vol. 12, Issue 5, May 2023.

In cloud storage, how to search sensitive data efficiently and securely is a challenging problem. The searchable encryption technique provides a secure storage method without loss of data confidentiality and usability. As an important branch of searchable encryption, public-key encryption with keyword search (PEKS) is widely studied by scholars. The main aim is to provide secure and efficient searching of encrypted data in a cloud environment. It enables a user to delegate the storage of their data to cloud servers while still maintaining control over the confidentiality of the data. It ensures that the data and the search queries are processed by two different servers. The user encrypts their data using a public key and uploads it to one server, while the search queries are sent to another server. This separation of data and search queries ensures that no single server has access to both the data and the search queries. The scope of DS-PKE-KWS is in the field of secure data storage and cloud computing, and it can be applied in various domains where sensitive data needs to be stored and protected while allowing authorized users to search for specific information without revealing any sensitive information to the cloud servers. The ability to interchange the roles of the AS and the TS also improves the efficiency of the scheme in practical applications. This module represents the encrypted data stored in the cloud storage. The data is encrypted using the DPAEKS scheme, which provides strong security and ensures data confidentiality. The Intermediate Cipher Text is stored in the cloud storage and is used for keyword search operations without revealing the plaintext data to the cloud storage provider or any other unauthorized entities. They have presented a new scheme called dual-server public-key authenticated encryption with keyword search (DPAEKS). The features of DPAEKS include: two non-colluding servers that are used to protect against IKGA and the data owner should be distributed with a pair of keys to authenticate the data. We developed a concrete construction of DPAEKS and proved its security. Finally, we implemented and evaluated the performance of the proposed scheme. The empirical results we obtained

demonstrate that it is suitable for deployment in practical applications. The performance evaluation of the proposed scheme may have demonstrated its suitability for practical applications, but future work can focus on further improving the scalability and efficiency of the scheme. This can involve optimizing the computational overhead, reducing communication overhead, and exploring techniques to handle larger keyword spaces or larger datasets to enhance the overall performance of the scheme in real-world scenarios.

**[5] Keyword-based Ciphertext Search Algorithm under Cloud Storage,
En Xunyi1, Issue 4, April 2022**

Cloud storage has the advantages of cost effective, high scalability, inexpensive, without access limit and easy to manage. The behavior of confidential data is saved to third parties caused the company's great worry, which limit the development of cloud storage. Many enterprises have to adopt the traditional storage model. This has become an important factor of restricting the development of cloud storage. Under these conditions, encrypted storage produced. In the conditions of large amount of ciphertext data stored in server-side ,users need to put all of the ciphertext data downloaded to the local and retrieve after decrypting data. Although this strategy protects the privacy of data, but users need to waste a lot of resources, consume large amounts of network bandwidth, which also poses a serious obstacle to the development of cloud storage. At the same time, a huge cloud server computing capacity were idle, and became a simple data pool. In order to exert cloud server computing power and huge storage resources, keyword-based search technologies emerged. After encrypting, client upload documents and inverted index to the cloud server. When searching, users only need to enter a keyword. Cloud Server can search for relevant ciphertext file and then return to the client. Local users can use the key to decrypt the ciphertext file. This program not only protect the security of data, but also use computing power and large storage resources. Files use symmetric key encryption algorithm to encrypt and inverted indices also use a pseudo-random function to encrypt. Ciphertext and the encrypted indices upload to the server with storage. When the user queries, client hosts use the search algorithm to process the keyword to a query pointer and submit to the server. The server uses the pointer to call the user's encrypted search index. Server use the user's encrypted index to find the corresponding ciphertext and returns to the client, the client can use the key to decrypt the ciphertext. With this idea, the cloud server will not get any useful information about files. Server just know the ciphertext, encrypted index, and search pointer

and cannot get any effective plaintext and keyword information. Through the above analysis, programme extract keywords from user's files, and transform plaintext files and keywords to ciphertext and encrypted index respectively. These two steps can be done on the client. Ciphertext and encrypted indices are stored on cloud server. When the user queries, client hosts use the search algorithm to process the keyword to a query pointer and submit to the server. The server uses the pointer to call the user's encrypted search index. Server use the user's encrypted index to find the corresponding ciphertext and returns to the client, the client can use the key to decrypt the ciphertext.

CHAPTER 3

3.ANALYSIS

3.1 PROBLEM STATEMENT:

Symmetric key encryption for cloud storage security

(Building an effective security mechanism for storing documents)

3.2 EXISTING SYSTEMS:

Existing cloud security software are Trend micro one, cisco cloudlock, PaloAlto networks prisma cloud and soon. Trend micro one has the agent based security on Desktop/Laptop and error reporting is very difficult to maintain. Support is very slow to respond and resolve issues. Intrusion detection and prevention systems (IDPS) are among the most effective security tools on the market. They monitor, analyze, and respond to network traffic, either as a standalone solution or part of another tool that helps secure a network like a firewall. Major cloud services like Amazon, Azure and Google Cloud offer their own IDPS and firewall services for an additional cost. organizations should have written guidelines that specify who can use cloud services, how they can use them, and which data can be stored in the cloud. They also need to lay out the specific security technologies that employees must use to protect data and applications in the cloud.

Many organizations implement a defense-in-depth plan that includes:

- Firewalls
- Anti-malware
- Intrusion detection
- Access control

A cloud access security broker (CASB) is an on-premises or cloud-based security policy enforcement point that is placed between cloud service consumers and cloud service providers to combine and interject enterprise security policies as cloud-based resources are accessed. One potential challenge with CASB is the ability to integrate the solution with existing network infrastructure. Because of this, it can be difficult to manage security policies across multiple cloud services and applications within your infrastructure. Further, it can be difficult to define consistent policies, maintain compliance, and adapt to changing

cloud environments because they require ongoing attention and expertise over the entire network.

Cloud environments are vulnerable to network-based attacks, as tactics like distributed denial-of-service (DDoS) can throttle speed, block traffic, and cause unpredictable downtime. Unauthorized access can take longer to detect on the cloud than on physical, on-premise systems. If companies hand over ownership to a managed cloud services provider, there can be a significant delay in reporting and addressing the breach.

3.3 SYMMETRIC-KEY ENCRYPTION

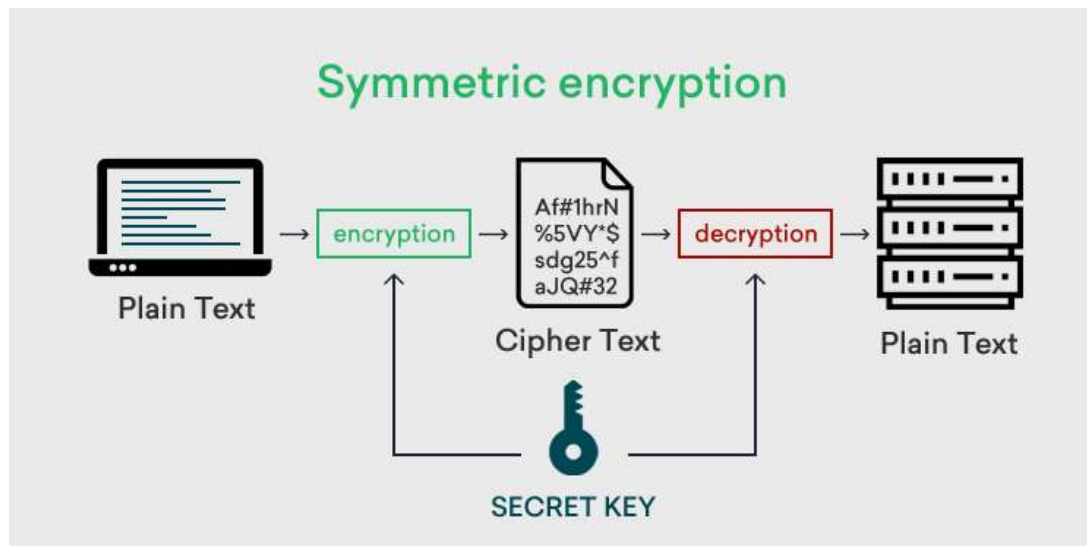


Fig: 3.1 Symmetric-Key Encryption

Working on Symmetric Encryption

These operations are performed to share the message securely over the network using the symmetric encryption technique.

Key Generation:

This is the first step in the symmetric encryption technique in which the private key needs to be chosen and must be securely communicated/ transferred over the network for the further use.

Encryption:

In this step, the plaintext (this is the original message to be sent over the network) is converted to some bogus, unintelligible text called the **ciphertext** using the shared secret key and the some algorithm.

Transfer of CipherText:

In this step the ciphertext is transferred over the network, since we have encrypted the original message even if this ciphertext is intercepted it will be unintelligible to the interceptor unless and until our shared secret key and algorithm is also compromised.

Decryption:

This is the last step where the receiver uses the reverse encryption algorithm and the shared secret key to convert the ciphertext back to the plaintext this is called decryption.

3.4 PROPOSED SYSTEM:

The simplest form of encryption is symmetric key encryption. This kind of encryption is also called private key encryption. With symmetric-key encryption, a single secret key can both lock the information and unlock the information. The user encrypts data with a private key that the user does not reveal to anyone else. If the key is a good one, no one else can decrypt the data.

1.speed and efficiency:

One of the main advantages of symmetric encryption algorithms is that they are fast and efficient. They require less computational power and memory than asymmetric encryption algorithms, which use different keys for encryption and decryption. This means that symmetric encryption algorithms can process large amounts of data quickly and with minimal overhead. For example, AES, a popular symmetric encryption algorithm, can encrypt and decrypt data at a rate of several gigabytes per second on modern hardware.

2 Simplicity and compatibility

Another advantage of symmetric encryption algorithms is that they are simple and compatible. They have been around for a long time and have been standardized and tested by various organizations and authorities. They are also widely supported by different

platforms, devices, and protocols. For example, symmetric encryption algorithms are used in SSL/TLS, the protocol that secures web traffic, and in WPA2, the protocol that secures wireless networks. Symmetric encryption algorithms are also easier to implement and understand than asymmetric encryption algorithms, which involve complex mathematical concepts and operations.

3 Security and confidentiality

A third advantage of symmetric encryption algorithms is that they provide a high level of security and confidentiality. They use strong keys that are hard to guess or crack by brute force attacks. They also use various techniques, such as padding, block chaining, and modes of operation, to prevent common attacks, such as replay, modification, and analysis. For example, AES, which uses a 128-bit, 192-bit, or 256-bit key, is considered to be secure against any known attacks and is approved by the US government for classified information. With the help of symmetric key encryption process using AES algorithm, the documents can be encrypted securely and stored in the cloud. The main pillars of encryption called confidentiality, integrity and availability can be thus proved.

CHAPTER 4

4. DESIGN

Software design has historically focused on developing code to provide desired or required functionality. It is defined as the entire process of defining an overall structure, such as software methods, functions, objects, and interface of your code to get noteworthy functionality. Any complex system is best understood by making some kind of diagrams or pictures. These diagrams have a better impact on our understanding. We prepare UML diagrams to understand the system in a better and simple way.

UML, which stands for Unified modelling language, is a way to visually represent the architecture, design, and implementation of complex software systems. When you're writing code, there are thousands of lines in an application, and it's difficult to keep track of the relationships and hierarchies within a software system. UML diagrams divide that software system into components and subcomponents. UML diagrams define a standard way to visualize the entire system of software.

Why should you use UML diagrams?

- UML is a standardized modeling language that can be used across different programming languages and development processes, so the majority of software developers will understand it and be able to apply it to their work.
- Bring new team members or developers switching teams up to speed quickly.
- Navigate to source code.
- Plan out new features before any programming takes place.
- Communicate with technical and non-technical audiences more easily.

4.1 UML DESIGN

- Class Diagram
- Activity Diagram
- Use-Case Diagram
- Sequence Diagram

- Communication Diagram
- Component Diagram
- Deployment Diagram

4.1.1 Class Diagram:

Class diagrams are the main building block of any object-oriented solution. It shows the classes in a system, attributes, and operations of each class and the relationship between each class. It depicts the static structure of the system. In most modeling tools, a class has three parts. Name at the top, attributes in the middle and operations or methods at the bottom.

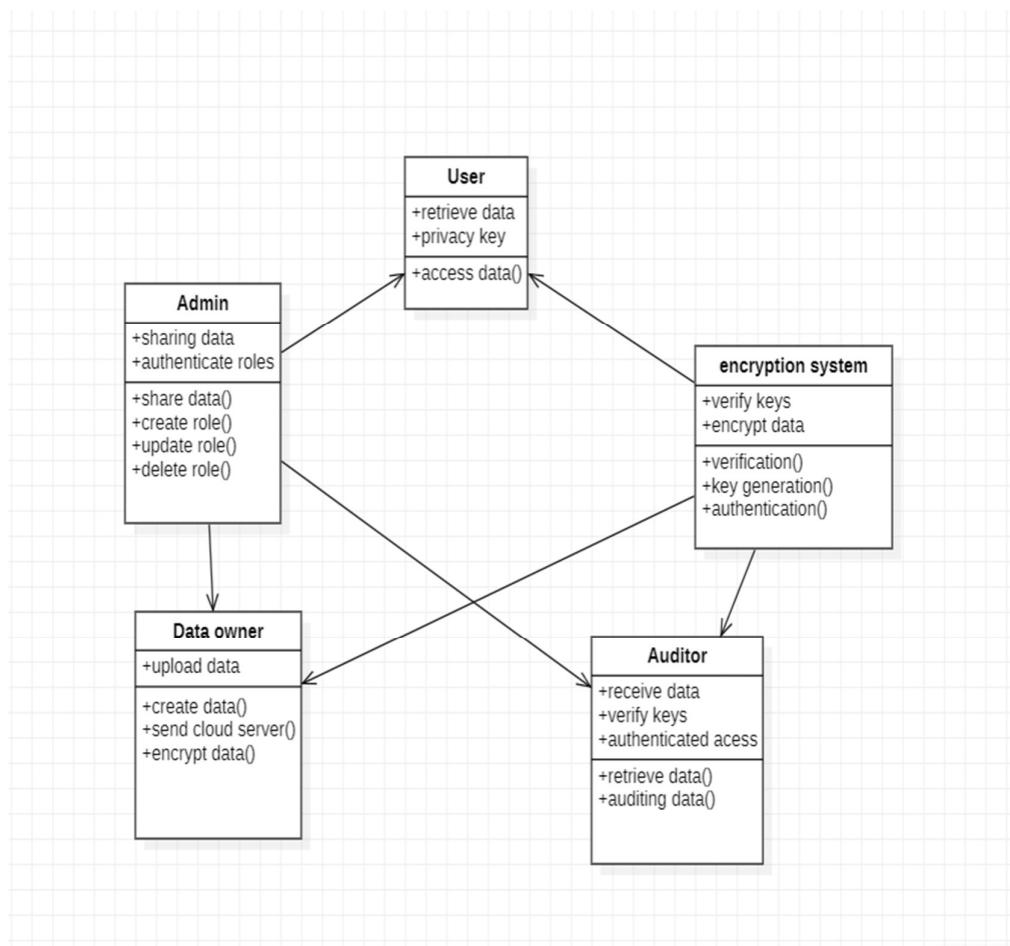


Fig: 4.1

4.1.2 Activity Diagram:

Activity diagrams represent workflows in a graphical way. They can be used to describe the business workflow or the operational workflow of any component in a system. It models the flow of control from one activity to the other. With the help of an activity diagram, we can model sequential and concurrent activities. It visually depicts the workflow as well as what causes an event to occur.

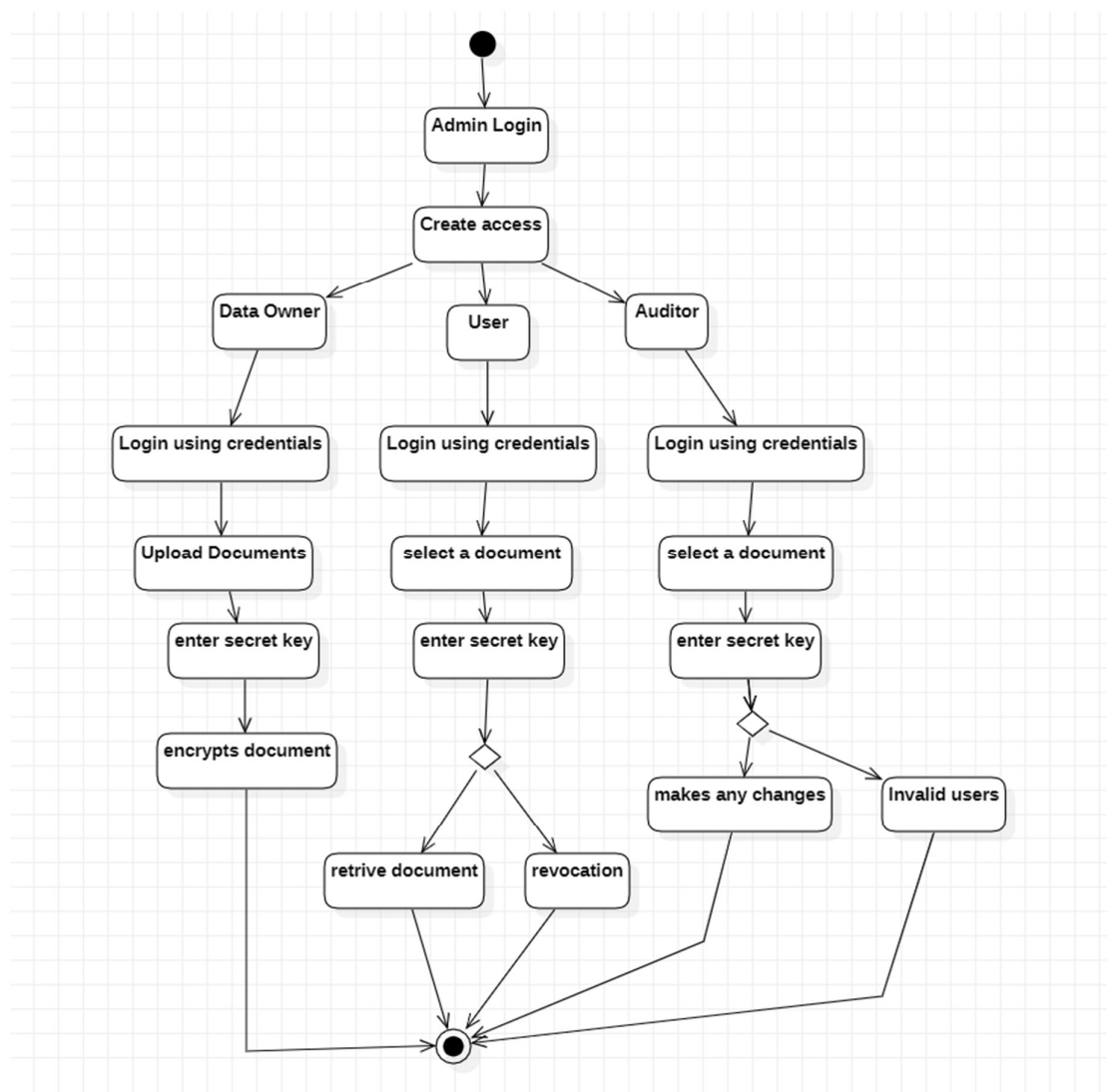


Fig: 4.2

4.1.3 Use Case Diagram:

The purpose of a use case diagram in UML is to demonstrate the different ways that a user might interact with a system. A use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system.

UML use case diagrams are ideal for:

- ✓ Representing the goals of system-user interactions
- ✓ Defining and organizing functional requirements in a system
- ✓ Specifying the context and requirements of a system
- ✓ Modeling the basic flow of events in a use case.

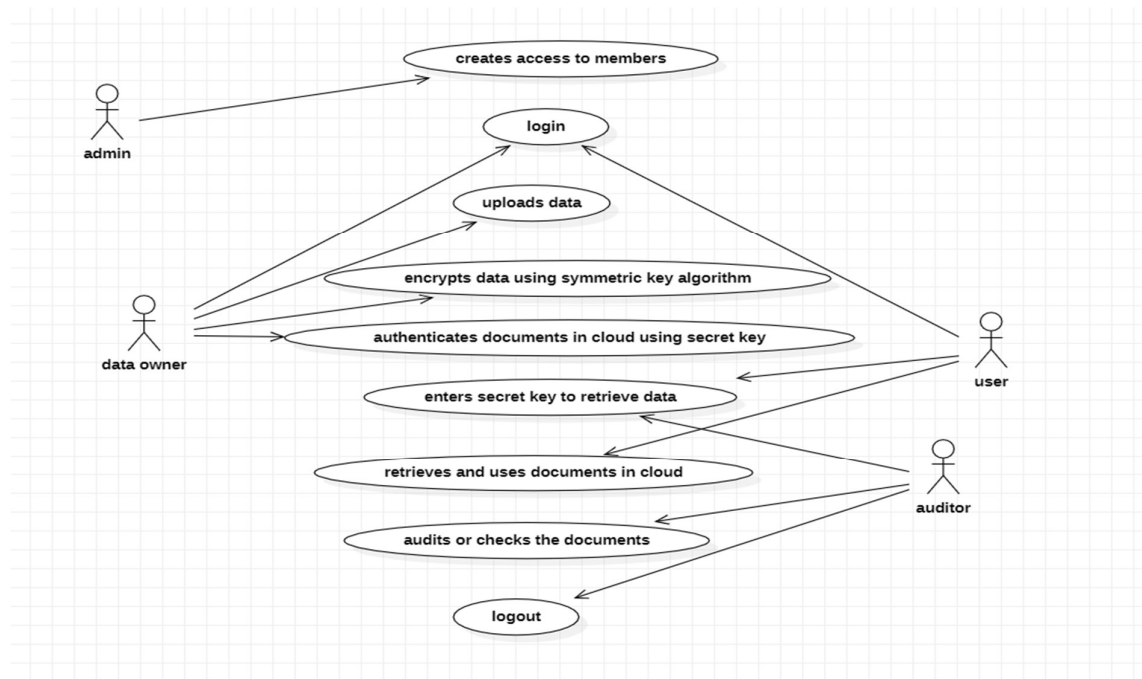


Fig: 4.3

4.1.4 Sequence Diagram:

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios. These diagrams examine how objects and components interact with each other to complete a process.

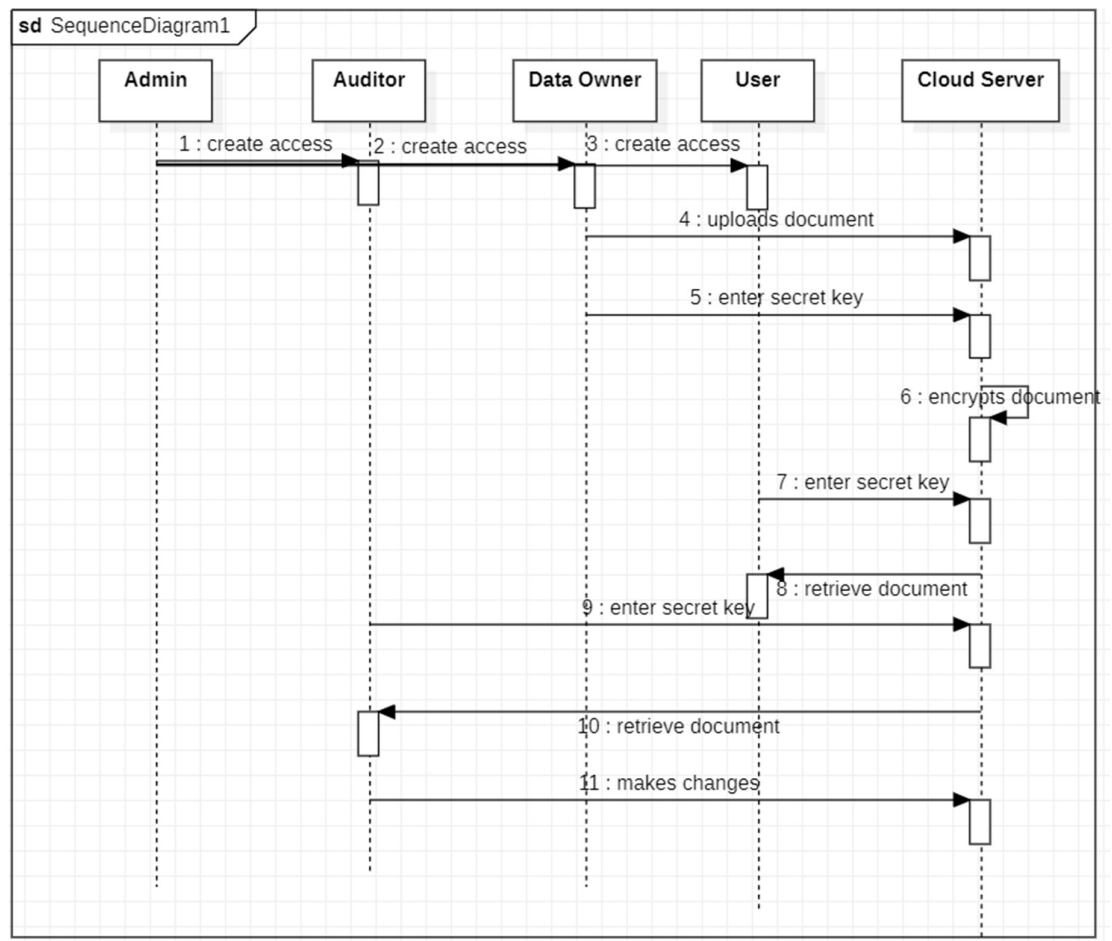


Fig: 4.4

4.1.5 Communication Diagram:

A communication diagram offers the same information as sequence diagram, but a sequence diagram emphasizes the time and order of events, a communication diagram emphasizes the messages exchanged between objects in an application. They are used to identify how commands are sent and received between objects or components of a process. Communication diagrams provide a scope to plan and understand the detailed functionality of an existing or future scenario.

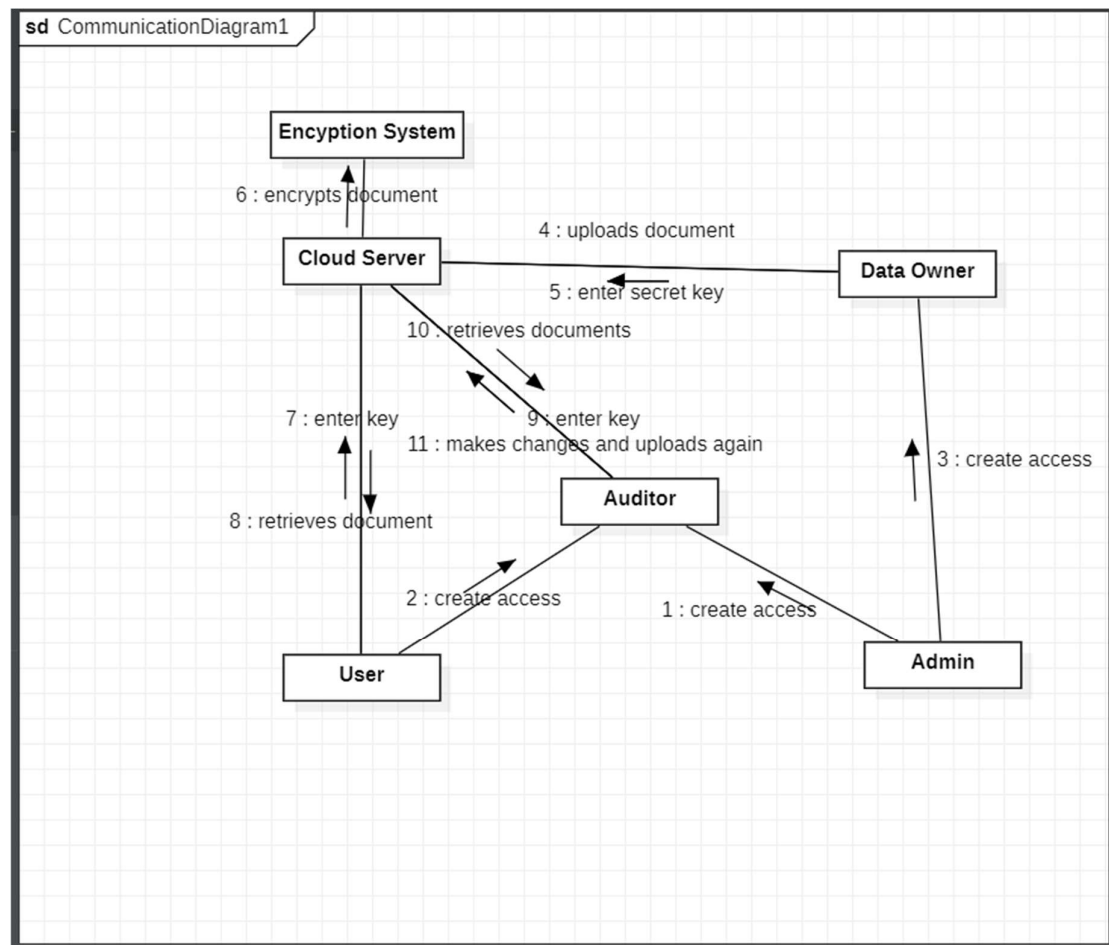


Fig: 4.5

4.1.6 Component Diagram:

A component diagram breaks down the actual system under development into various high levels of functionality. Each component is responsible for one clear aim within the entire system and only interacts with other essential elements on a need-to-know basis. A component represents a modular part of a system that encapsulates its contents and whose manifestation is replaceable within its environment.

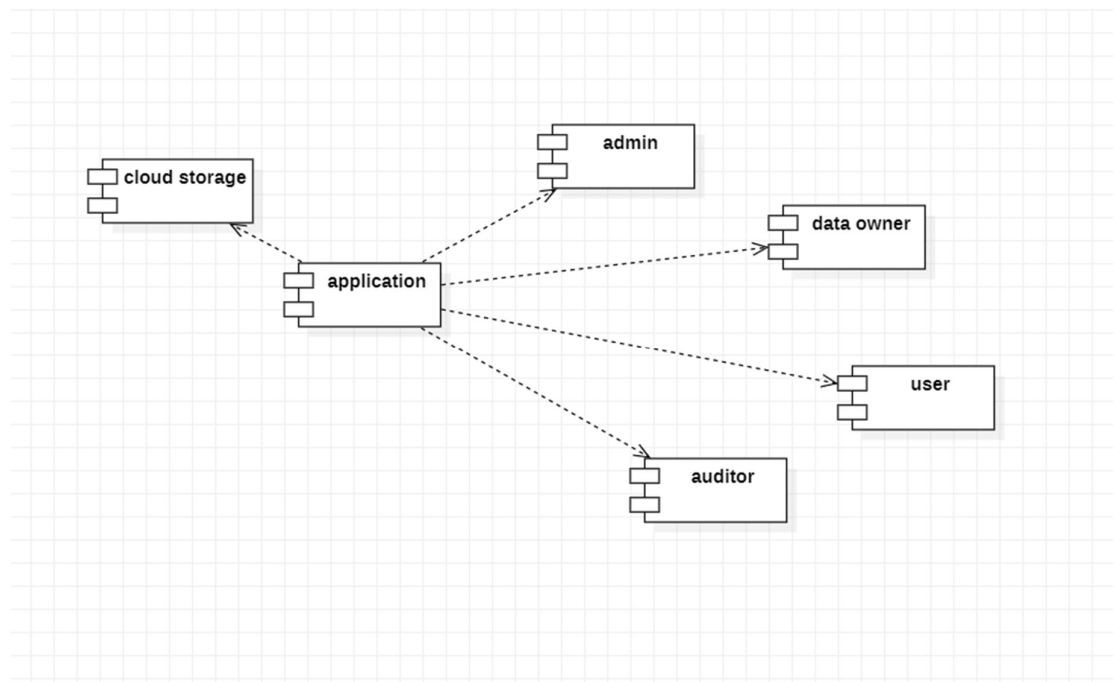


Fig: 4.6

4.1.7 Deployment Diagram:

A UML deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modelling the physical aspects of an object-oriented system. They are often be used to model the static deployment view of a system (topology of the hardware).

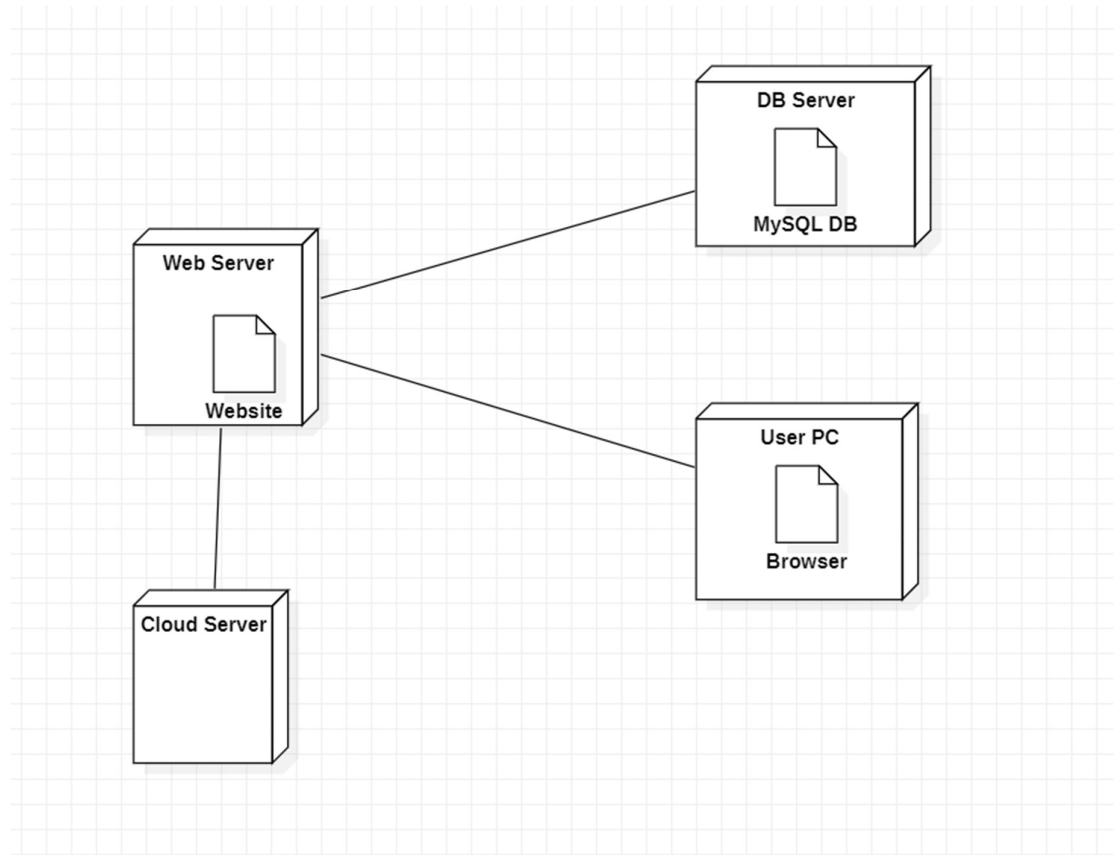


Fig: 4.7

4.2 System Architecture

System architecture serves as the blueprint for the design, implementation, and maintenance of complex software projects. It defines the structure, components, interactions, and principles that govern how the system will function and evolve over time. The importance of system architecture in a project cannot be overstated, as it influences every aspect of the development lifecycle and directly impacts the success and longevity of the software product.

System architecture provides the framework within which all software components interact and operate. It defines the high-level structure of the system, including its modules, layers, interfaces, and dependencies. By establishing this framework early in the project lifecycle, architects set the stage for efficient development, testing, and deployment of the software.

Its careful design and implementation are essential for aligning with project objectives, enabling scalability and flexibility, enhancing performance and efficiency, ensuring reliability and resilience, facilitating collaboration and communication, and supporting maintenance and evolution.

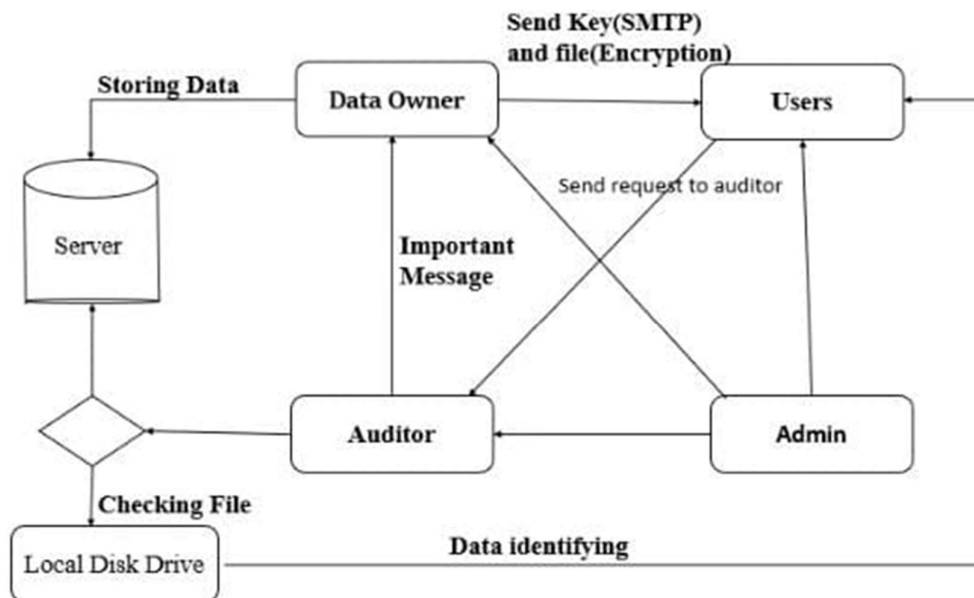


Fig: 4.8

CHAPTER 5

5. IMPLEMENTATION

5.1 JAVA:

In the ever-evolving landscape of computer programming, Java stands out as a versatile and robust language that has left an indelible mark on the world of software development. From web applications to mobile apps and enterprise systems, Java has become a cornerstone of modern computing, empowering developers to build powerful, scalable, and secure solutions for a diverse range of applications.

Features and Characteristics:

One of the defining features of Java is its platform independence. Java programs are compiled into bytecode, which can run on any device or platform that has a Java Virtual Machine (JVM). This portability has made Java the language of choice for cross-platform development, enabling developers to write code once and run it anywhere.

Java's object-oriented nature promotes code reuse, modularity, and maintainability.

Security is another key aspect of Java's design. With built-in features such as sandboxing and bytecode verification, Java provides a secure environment for executing code, protecting against unauthorized access and malicious attacks.

Applications and Use Cases: Java's versatility is reflected in its wide range of applications. It is commonly used for web development, with frameworks like Spring and Java Server Faces (JSF) enabling the creation of dynamic and interactive web applications. Java is also a popular choice for mobile app development, powering the Android platform and millions of mobile devices worldwide.

In the enterprise world, Java is the language of choice for building large-scale, mission-critical systems. From customer relationship management (CRM) to enterprise resource planning (ERP) and financial systems, Java provides the scalability, reliability, and performance required for enterprise-grade applications.

Significance and Future: The significance of Java in the world of programming cannot be overstated. As technology continues to evolve, Java remains at the forefront, adapting to new challenges and opportunities while staying true to its core principles of simplicity, portability, and security.

In conclusion, Java is more than just a programming language – it is a cornerstone of modern computing, powering the digital age and driving innovation across industries. With its rich features, wide range of applications, and unwavering commitment to excellence, Java continues to inspire and empower the next generation of software developers, ensuring that its legacy will endure for years to come.

5.2 JSP

Java Server Pages (JSP): Unleashing Dynamic Web Development Java Server Pages (JSP) is a powerful technology in the realm of web development, empowering developers to create dynamic and interactive web content using Java.

Uses of JSP:

Dynamic Content Generation: JSP enables the creation of web pages with dynamic content, allowing developers to embed Java code directly into HTML pages. This facilitates the generation of personalized content based on user input, database queries, or other external factors.

Server-side Processing: JSP runs on the server-side, enabling server-side processing of user requests. This allows for the execution of complex business logic, database interactions, and session management, enhancing the interactivity and responsiveness of web applications.

Advantages of JSP: JavaServer Pages (JSP) offers several advantages, including:

Ease of Development: JSP simplifies web development by allowing developers to use familiar HTML syntax along with embedded Java code. This reduces the learning curve for developers and accelerates the development process.

Dynamic Content Generation: With JSP, developers can generate dynamic content on-the-fly, enabling the creation of interactive web pages that respond to user input in real-time.

Reusable Components: JSP promotes the creation of reusable components through custom tags and JavaBeans, facilitating code reuse, modularity, and maintainability.

MVC Architecture Support: JSP follows the Model-View-Controller (MVC) architecture, separating the presentation layer (view) from the business logic (model) and application flow control (controller). This promotes code organization, scalability, and maintainability.

Portability: JavaServer Pages (JSP) applications are platform-independent, allowing them to run on any server that supports Java and a Java EE container.

Applications of JSP:

Java Server Pages (JSP) finds applications in various domains, including:

Dynamic Websites: JSP is commonly used for creating dynamic websites that display personalized content, such as e-commerce platforms, social networking sites, and news portals.

Web Applications: JSP is a key technology for developing web-based applications that require server-side processing, such as online banking systems, customer relationship management (CRM) software, and enterprise resource planning (ERP) systems.

In conclusion, JavaServer Pages (JSP) is a versatile and powerful technology for creating dynamic and interactive web content using Java. With its ease of development, dynamic content generation capabilities, support for reusable components, adherence to MVC architecture, and wide range of applications, JSP continues to be a popular choice for web developers seeking to build robust and scalable web applications.

5.3 CSS

In the vast landscape of web development, Cascading Style Sheets (CSS) emerge as a cornerstone technology, playing a pivotal role in defining the visual presentation and layout of web pages. From simple styling to complex design transformations, CSS empowers developers to create visually stunning and user-friendly websites.

Features of CSS:

Separation of Content and Presentation: CSS allows web developers to separate the content (HTML structure) from its presentation (styling). This separation promotes clean code organization, enhances maintainability, and facilitates design changes without altering the underlying content.

Selector-Based Styling: CSS enables developers to apply styles to HTML elements using selectors, which target specific elements based on their attributes, IDs, classes, or relationships with other elements.

Cascade and Specificity: CSS follows the cascade and specificity rules, which determine the order of precedence for applying styles to elements.

Responsive Design: CSS facilitates the creation of responsive web designs that adapt to different screen sizes and devices.

Advantages of CSS:

Enhanced User Experience: CSS enables developers to create visually appealing and user-friendly websites that enhance the overall user experience. Well-designed layouts, typography, colors, and visual elements contribute to improved usability and engagement.

Improved Accessibility: CSS allows developers to enhance the accessibility of web content by providing semantic markup and ensuring proper contrast, font sizes, and spacing.

Efficiency and Consistency: CSS promotes code efficiency and consistency by enabling developers to define styles once and apply them across multiple elements or pages. This reduces redundancy, simplifies maintenance, and ensures a consistent look and feel across the entire website.

Objectives of CSS:

Modularity: CSS aims to promote modularity by enabling developers to create reusable stylesheets that can be applied across multiple web pages. This modularity enhances code maintainability, scalability, and consistency.

Accessibility: CSS aims to enhance the accessibility of web content by providing semantic markup, proper document structure, and styling that accommodates different user needs and preferences. Accessible designs ensure that all users can access and interact with web content effectively.

Applications of CSS:

Website Design: CSS is widely used for designing websites of all types, including personal blogs, corporate websites, e-commerce platforms, and portfolio sites. Its styling capabilities enable developers to create visually appealing and user-friendly layouts that resonate with target audiences.

Web Application Development: CSS is essential for developing web applications that require rich and interactive user interfaces.

In conclusion, Cascading Style Sheets (CSS) are fundamental to modern web development, providing developers with the tools and techniques needed to create visually stunning, user-friendly, and accessible websites and web applications. With its features, advantages, objectives, and diverse applications, CSS continues to shape the web and elevate the online experience for users worldwide.

Symmetric key encryption, a cornerstone of modern cryptography, finds extensive applications across various domains, where confidentiality, integrity, and authenticity of data are paramount. This essay explores the multifaceted applications of symmetric key encryption and its crucial role in safeguarding sensitive information in today's interconnected world.

Symmetric key encryption is widely employed to secure data stored on various devices and platforms, including computers, servers, mobile devices, and cloud storage services. By encrypting data at rest, symmetric encryption protects it from unauthorized access in case of theft, loss, or data breaches. This application is particularly crucial for sensitive information such as financial records, healthcare data, and personal documents, where confidentiality is paramount.

5.4 AES Algorithm:

Advanced encryption algorithm(AES) is a specification for the encryption of electronic data established by the U.S National Institute of Standards and Technology (NIST) in 2001. AES is widely used today as it is a much stronger than DES and triple DES despite being harder to implement.

- AES is a block cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each.

That means it takes 128 bits as input and outputs 128 bits of encrypted cipher text as output. AES relies on substitution-permutation network principle which means it is performed using a series of linked operations which involves replacing and shuffling of the input data.

Working of AES Algorithm

AES performs operations on bytes of data rather than in bits. Since the block size is 128 bits, the cipher processes 128 bits (or 16 bytes) of the input data at a time.

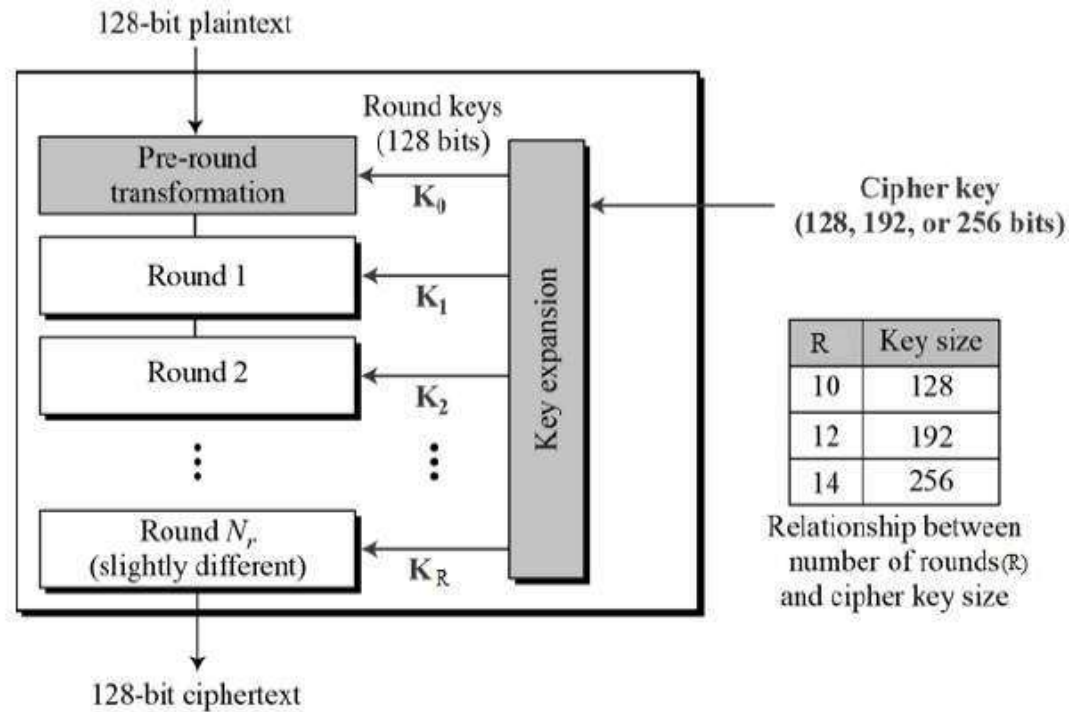


Fig: 5.1

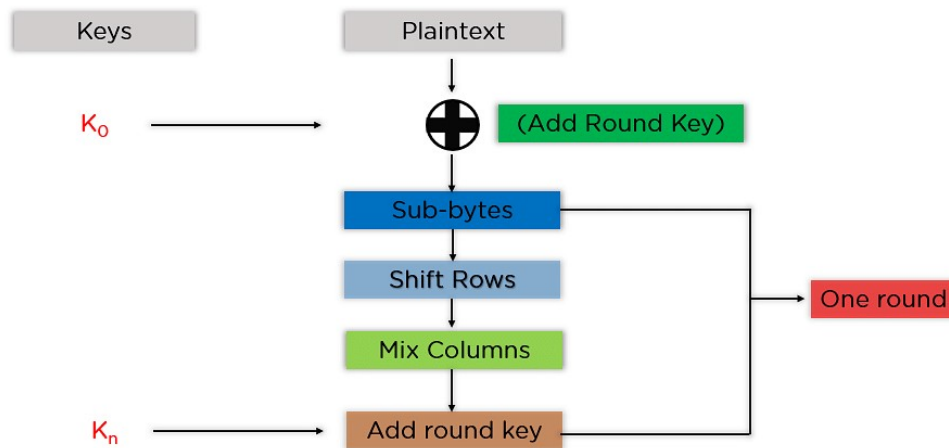


Fig: 5.2

Add Round Key: You pass the block data stored in the state array through an XOR function with the first key generated (K_0). It passes the resultant state array on as input to the next step.

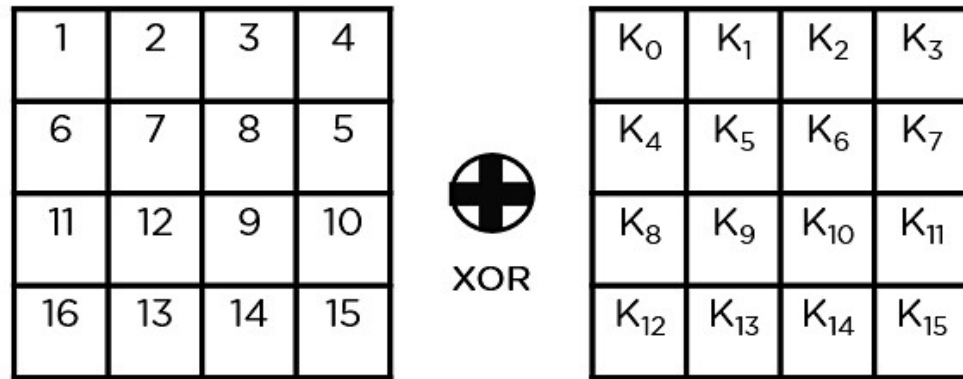


Fig: 5.3

SubBytes:

In this step, it converts each byte of the state array into hexadecimal, divided into two equal parts. These parts are the rows and columns, mapped with a substitution box (S-Box) to generate new values for the final state array. This step implements the substitution.

- In this step each byte is substituted by another byte. Its performed using a lookup table also called the S-box. This substitution is done in a way that a byte is never substituted by itself and also not substituted by another byte which is a compliment of the current byte. The result of this step is a 16 byte (4 x 4) matrix like before.

The next two steps implement the permutation.

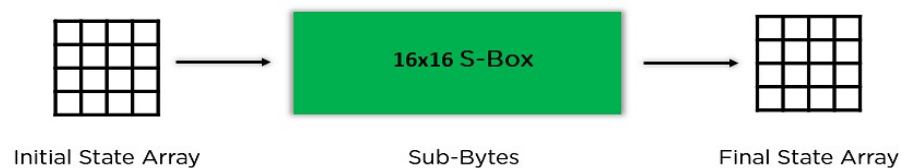


Fig: 5.4

ShiftRows:

It swaps the row elements among each other. It skips the first row. It shifts the elements in the second row, one position to the left. It also shifts the elements from the third row two consecutive positions to the left, and it shifts the last row three positions to the left.

This step is just as it sounds. Each row is shifted a particular number of times.

- The first row is not shifted
- The second row is shifted once to the left.
- The third row is shifted twice to the left.
- The fourth row is shifted thrice to the left.

(A left circular shift is performed.)

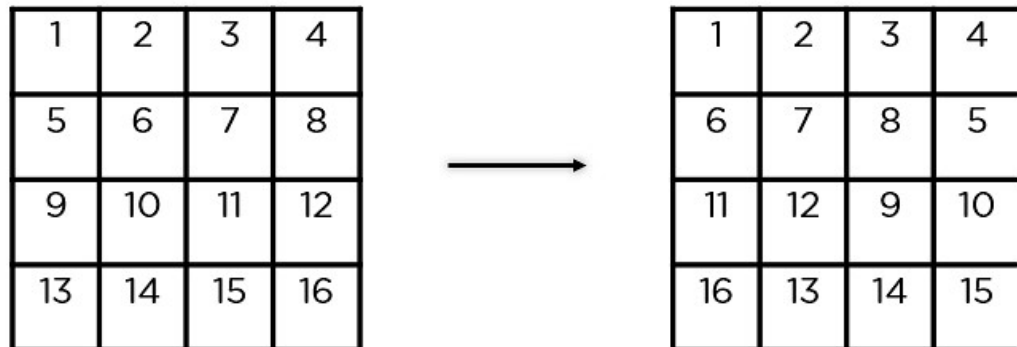


Fig: 5.5

MixColumns:

This step is basically a matrix multiplication. Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result.

This step is skipped in the last round.

The last round doesn't have the MixColumns round.

The SubBytes does the substitution and ShiftRows and MixColumns performs the permutation in the algorithm.

Decryption:

The stages in the rounds can be easily undone as these stages have an opposite to it which when performed reverts the changes. Each 128 blocks goes through the 10, 12 or 14 rounds depending on the key size.

The stages of each round in decryption is as follows :

- Add round key
- Inverse MixColumns
- ShiftRows
- Inverse SubByte

The decryption process is the encryption process done in reverse.

5.5 Implementation of AES Algorithm:

```
13 import javax.crypto.Cipher;
14 import javax.crypto.SecretKey;
15 import javax.crypto.spec.IvParameterSpec;
16 import javax.crypto.CipherInputStream;
17 import javax.crypto.CipherOutputStream;
18 import javax.crypto.KeyGenerator;
19
20 import java.security.spec.AlgorithmParameterSpec;
21 import javax.crypto.spec.SecretKeySpec;
22
23 public class AESEncrypter
24 {
25     Cipher ecipher;
26     Cipher dcipher;
27
28     public AESEncrypter(SecretKey key)
29     {
30         // Create an 8-byte initialization vector
31         byte[] iv = new byte[]
32         {
33             0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f
34         };
35
36         AlgorithmParameterSpec paramSpec = new IvParameterSpec(iv);
37         try
38         {
39             ecipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
40             dcipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
41
42             // CBC requires an initialization vector
43             ecipher.init(Cipher.ENCRYPT_MODE, key, paramSpec);
44             dcipher.init(Cipher.DECRYPT_MODE, key, paramSpec);
45         }
46         catch (Exception e)
47         {
48             e.printStackTrace();
49         }
50     }
51
52     // Buffer used to transport the bytes from one stream to another
53     byte[] buf = new byte[1024];
54 }
```

Fig: 5.6


```

52 // Buffer used to transport the bytes from one stream to another
53 byte[] buf = new byte[1024];
54
55 public void encrypt(InputStream in, OutputStream out)
56 {
57     try
58     {
59         // Bytes written to out will be encrypted
60         out = new CipherOutputStream(out, ecipher);
61
62         // Read in the cleartext bytes and write to out to encrypt
63         int numRead = 0;
64         while ((numRead = in.read(buf)) >= 0)
65         {
66             out.write(buf, 0, numRead);
67         }
68         out.close();
69     }
70     catch (java.io.IOException e)
71     {
72     }
73 }
74
75
76 public void decrypt(InputStream in, OutputStream out)
77 {
78     try
79     {
80
81         // Bytes read from in will be decrypted
82         in = new CipherInputStream(in, dcipher);
83
84         // Read in the decrypted bytes and write the cleartext to out
85         int numRead = 0;
86         while ((numRead = in.read(buf)) >= 0)
87         {
88             out.write(buf, 0, numRead);
89         }
90         out.close();
91     }
92     catch (java.io.IOException e)
93     {

```

Fig: 5.7

CHAPTER 6

6. TESTING

6.1 Testing the frontend and backend:

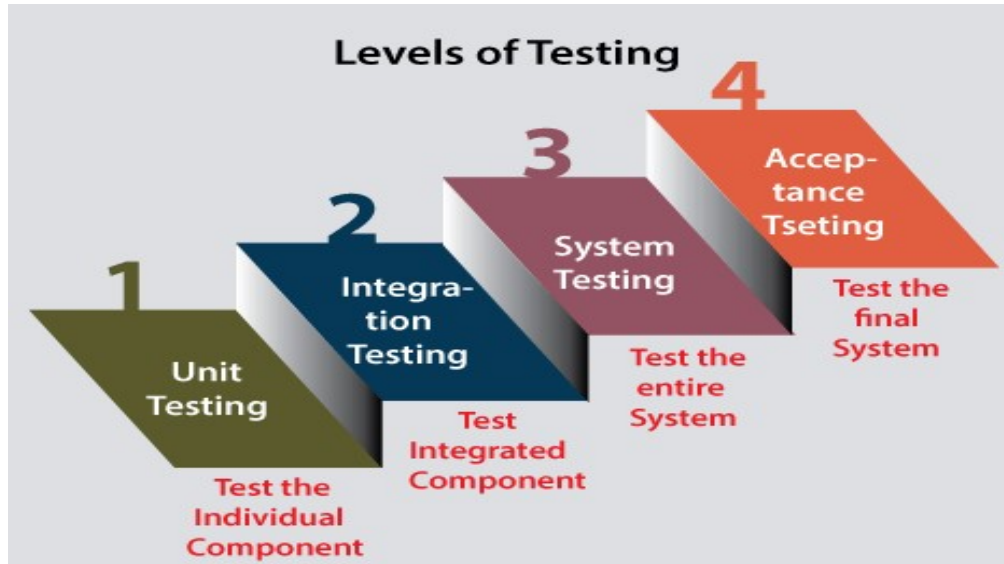


Fig: 6.1

Testing frontend and backend of an application involved various types of testing methodologies to ensure that both components function correctly and interact seamlessly.

The whole testing of frontend and backend was done manually and no automated tools were used.

6.1.1 Unit Testing:

Frontend: Unit testing for the frontend involved testing individual components, such as UI elements, functions, and modules, in isolation.

Backend: Unit testing for the backend focused on testing individual functions, classes, and APIs without dependencies on external systems or databases.

6.1.2 Integration Testing:

Frontend: Integration testing for the frontend involved testing the interaction between different components, modules, or pages to ensure they work together as expected. This included testing UI navigation, data flow, and component communication.

Backend: Integration testing for the backend involved testing the interaction between various backend components, such as APIs, databases, and external services. This ensures that the backend system behaves correctly as a whole.

6.1.3 System Testing:

Frontend: System testing for the frontend involved testing the entire application from the user's perspective. This includes testing UI functionalities, user interactions, accessibility, and responsiveness across different browsers and devices.

Backend: System testing for the backend involved testing the complete application stack, including frontend-backend interactions, data flow, security, performance, and scalability.

6.1.4 Validation Testing:

Frontend: Validation testing for the frontend involves verifying that the application meets the specified requirements and expectations.

Backend: Validation testing for the backend involved confirming that the backend system meets the functional and non-functional requirements outlined during the development process. This included testing data integrity, security controls, compliance with industry standards, and performance benchmarks.

6.2 Testing the working of secret key:

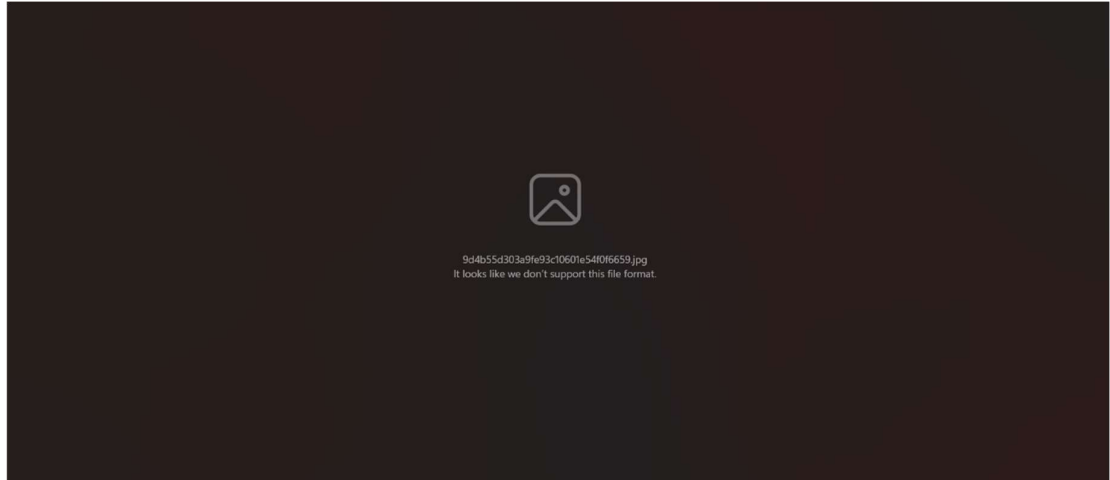


Fig: 6.2

The secret key entered by the user, if incorrect then it will not display the original picture. In order to keep the cloud authenticated, integrated the secret key has to be entered correctly, for avoiding any unauthenticated parties to retrieve the information.

CHAPTER 7

7. RESULTS



Fig: 7.1 Main page

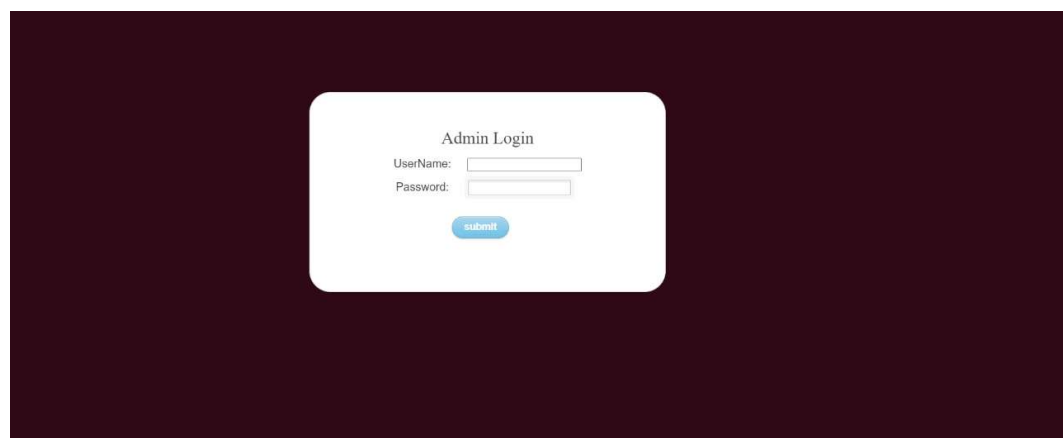



Fig: 7.2 Admin login page



Fig: 7.3 Registration page

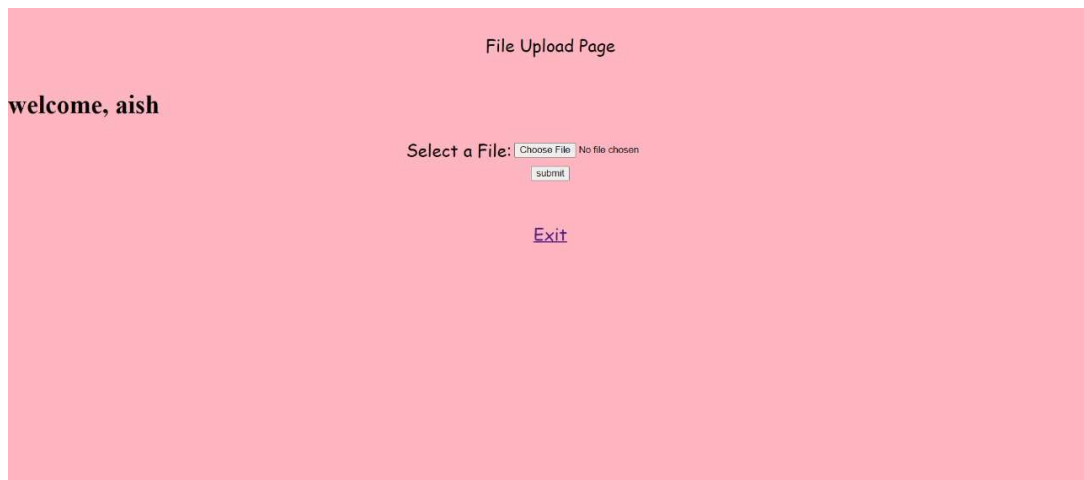
A screenshot of a registration form titled "Registration Form" with a yellow pushpin icon. The form is centered on a white background, flanked by dark red vertical bars. Below the title, the instruction "Please Complete the Fields Below:-" is shown. The form contains seven input fields, each with a label to its left: "UserName:", "Password:", "ConfirmPassword:", "Email:", "ProductKey:", "Mobile:", and "City:". At the bottom left of the form is a button labeled "submit".

Fig: 7.4 Data Owner Registration Form



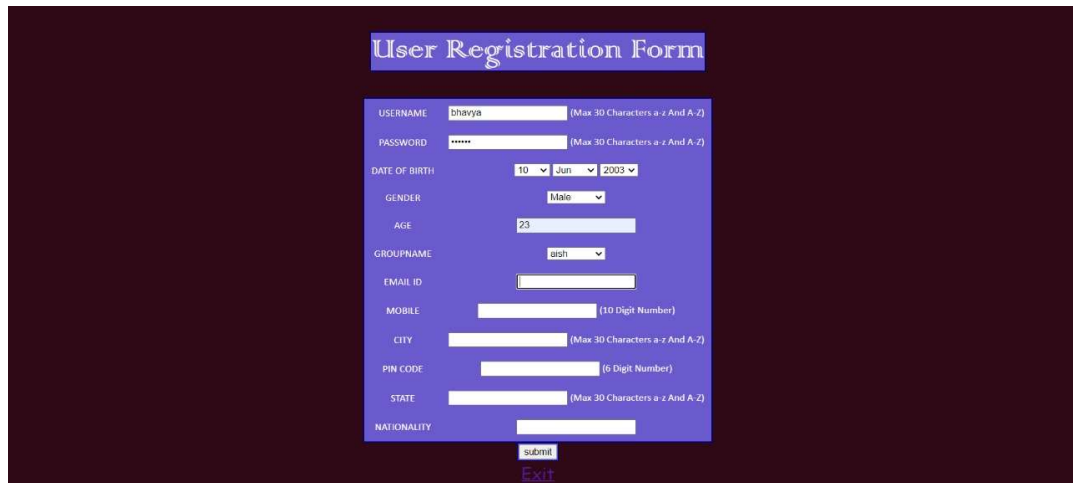
A login form titled "Login Form" is centered on a dark red background. The form has a dark gray header with the title. Below the header, there are two input fields: "UserName:" and "Password:". At the bottom of the form is a "submit" button.

Fig:7.5 Data Owner Login Form



A file upload page with a pink background. At the top, it says "File Upload Page". On the left, it says "welcome, aish". In the center, there is a "Select a File:" label followed by a file selection button that says "Choose File" and "No file chosen". Below this is a "submit" button. At the bottom, there is a blue "Exit" link.

Fig: 7.6 File Upload Page

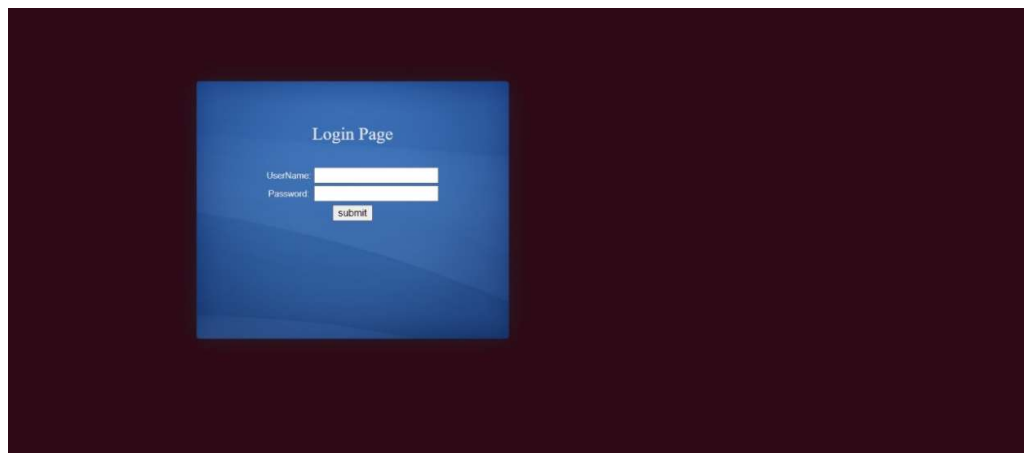


The image shows a 'User Registration Form' with a light blue header. The form fields are as follows:

Field	Value	Constraint
USERNAME	bhavya	(Max 30 Characters a-z And A-Z)
PASSWORD	*****	(Max 30 Characters a-z And A-Z)
DATE OF BIRTH	10 Jun 2003	
GENDER	Male	
AGE	23	
GROUPNAME	atish	
EMAIL ID		
MOBILE		(10 Digit Number)
CITY		(Max 30 Characters a-z And A-Z)
PIN CODE		(6 Digit Number)
STATE		(Max 30 Characters a-z And A-Z)
NATIONALITY		

At the bottom of the form are two buttons: 'submit' and 'Exit'.

Fig: 7.7 User Registration Page



The image shows a 'Login Page' with a blue gradient background. The form fields are as follows:

Field	Value
UserName:	
Password:	

Below the password field is a 'Submit' button.

Fig: 7.8 User Login Page

File Download Page

Select a File: 9d4b55d303a9f93c10801e54f0f8659.jpg ▼

Secret Key:

[Go to Home Page](#)

Fig: 7.9 File Download Page

CHAPTER 8

8.CONCLUSION

In the realm of cloud storage, where vast amounts of sensitive data are stored and transmitted across networks, security is paramount. Symmetric key encryption emerges as a robust solution for safeguarding data integrity, confidentiality, and authenticity in cloud storage environments. This essay explores the principles of symmetric key encryption and its application in securing cloud storage.

Symmetric key encryption, also known as secret key encryption, employs a single key for both encryption and decryption of data. Unlike asymmetric encryption, which involves a pair of keys (public and private), symmetric encryption simplifies the process by utilizing the same secret key for both encryption and decryption operations. This key is shared securely between the communicating parties or systems involved.

The application of symmetric key encryption in cloud storage begins with data encryption before transmission or storage. When a user uploads data to the cloud, it is encrypted using a symmetric encryption algorithm along with a secret key. This encrypted data is then transmitted to the cloud storage provider's servers or stored in cloud repositories. Since the encryption key is required for decryption, unauthorized access to the encrypted data yields no meaningful information.

One of the primary advantages of symmetric key encryption in cloud storage is its efficiency. Symmetric encryption algorithms, such as Advanced Encryption Standard (AES), are computationally efficient and can process large volumes of data swiftly. This efficiency is essential for cloud storage environments where rapid encryption and decryption of data are necessary to meet the demands of users accessing their data from various locations and devices. Moreover, symmetric key encryption ensures data confidentiality. Even if an unauthorized party gains access to the encrypted data stored in the cloud, they cannot decipher its contents without the corresponding secret key. This provides a layer of defense against data breaches and unauthorized disclosures, thereby bolstering user trust in cloud storage services.

In conclusion, symmetric key encryption serves as a fundamental building block for securing data in cloud storage. By encrypting data using a shared secret key, cloud storage providers can ensure confidentiality, integrity, and authenticity of stored data.

In modern-day enterprises, there has been a growing transition to cloud-based environments and IaaS, PaaS or SaaS computing models. The dynamic nature of infrastructure management, especially in scaling applications and services, can bring a number of challenges to enterprises when adequately resourcing their departments. These as-a-service models give organizations the ability to offload many of the time-consuming, IT-related tasks. By default, most cloud providers follow best security practices and take active steps to protect the integrity of their servers. However, organizations need to make their own considerations when protecting data, applications and workloads running on the cloud. Cloud security should be an important topic of discussion regardless of the size of your enterprise. Cloud infrastructure supports nearly all aspects of modern computing in all industries and across multiple verticals.

8.1 Future scope

- Integration with various standard cloud service platforms like Microsoft Azure, Amazon web services can be done for extensive storage and security.
- Mobile application can also be enabled for this system.
- Public key infrastructure(PKI) can also be integrated with this application.
- Elliptic curve cryptography may also be used.
- For better community engagement and collaboration by open-sourcing the project code, sharing in various platforms like GitHub and inviting contributions from various developers. This collaborative approach could accelerate innovation, sharing and facilitate developing new features for the project.

8.2 References

- [1] S. Kamara, C. Papamanthou and T. Roeder, "Dynamic searchable symmetric encryption," presented at ACM Conference on Computer and Communications Security, pp. 965-976, 2012.
- [2] C. Guo, X. Chen, Y. M. Jie, Z. J. Fu, M. C. Li and B. Feng, "Dynamic Multi-phrase Ranked Search over Encrypted Data with Symmetric Searchable Encryption," in IEEE Transactions on Services Computing, vol. 99, No. 1939, pp. 1-1, 2017.
- [3] Z. H. Xia, X. H. Wang, X. M. Sun and Q. Wang, "A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data," in IEEE Transactions on Parallel and Distributed Systems, vol. 27, No. 2, pp. 340-352.
- [4] S. Kamara and C. Papamanthou, "Parallel and Dynamic Searchable Symmetric Encryption," presented at the International Conference on Financial Cryptography and Data Security, pp. 258-274, 2013.
- [5] J. B. Yan, Y. Q. Zhang and X. F. Liu, "Secure multikeyword search supporting dynamic update and ranked retrieval," in China Communication, vol. 13, No. 10, pp. 209-221, 2016.
- [6] Lakhani, A. Gupta and K. Chandrasekaran, "IntelliSearch: A search engine based on Big Data analytics integrated with crowdsourcing and category-based search", International Conference on Circuits, Power and Computing Technologies , 2015.
- [7] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu. Plutus: Scalable Secure File Sharing on Untrusted Storage. In Proc. of FAST'03, Berkeley, California, USA, 2003.
- [8] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing remote untrusted storage," in Proc. of NDSS'03, 2003.
- [9] Dr. S. Vasundra et.al, CSE, JNTUACEA, Published a paper" Enabling Secure Data Sharing in the Cloud Storage Groups", International Research Journal of Engineering and Technology, ISSN: 2395-0056, July 2017.
- [10] Subashini S, Kavitha V (2011) A survey on security issues in service delivery models of cloud computing. J Netw Comput Appl 34:1–11.
- [11] Lu, Yue & Tan, Chew Lim., "Keyword searching in compressed document images". DCC, 2003..
- [12] S. S. Pawar, A. Manepatil, A. Kadam and P. Jagtap, "Keyword search in information retrieval and relational database system: Two class view," International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016.
- [13] Q. Dong, Z. Guan and Z. Chen, "Attribute-Based Keyword Search Efficiency Enhancement via an Online/Offline Approach," IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS), 2015.
- [14] Kehinde K. Agbele, Kehinde Daniel Aruleba, Eniafe F. Ayetiran, "Efficient schema based keyword search in relational databases." University of Computer Studies, Mandalay, Myanmar, International Journal of Computer Science, Engineering and Information Technology (IJCEIT) 2.6 (2012).