

A Major Project Report  
on  
**MULTI-AUTHORITY ATTRIBUTE BASED KEYWORD SEARCH OVER  
ENCRYPTED CLOUD DATA**  
submitted in partial fulfilment of the requirements for the award of the degree  
of  
**BACHELOR OF TECHNOLOGY**  
IN  
**COMPUTER SCIENCE AND ENGINEERING**  
BY

<b>LEKKALA SAI KETHANA</b>	<b>20EG105127</b>
<b>MADIPADIGE UDAY SAI KIRAN</b>	<b>20EG105128</b>
<b>N. MEGHAN SATWIK</b>	<b>20EG105134</b>
<b>POTTAPALLY KEERTHIPRIYA</b>	<b>20EG105141</b>

Under the guidance of

**Mr. MADAR BANDU**

Assistant Professor

Department of CSE



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Venkatapur(V), Ghatkesar(M), Medchal(D) - 500088**

**2023-2024**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

This is to certify that the report entitled “**MULTI-AUTHORITY ATTRIBUTE BASED KEYWORD SEARCH OVER ENCRYPTED CLOUD DATA**” that is being submitted by **LEKKALA SAI KETHANA [20EG105127]**, **MADIPADIGE UDAY SAI KIRAN [20EG105128]**, **N. MEGHAN SATWIK [20EG105134]**, **POTTAPALLY KEERTHIPRIYA [20EG105141]** in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering to the Anurag University, Hyderabad is a record of bonafide work carried out by them under my guidance and supervision. The results embodied in this Report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

**Internal Guide**

**Mr. Madar Bandu**

**Assistant Professor, Dept. of CSE**

**Dr. G. Vishnu Murthy**

**Professor & Dean, Dept. of CSE**

**External Examiner**

## ACKNOWLEDGMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **Mr. Madar Bandu** for his constant encouragement and inspiring guidance without which this project could not have been completed. His critical reviews and constructive comments improved our grasp of the subject and steered to the fruitful completion of the work. His patience, guidance and encouragement made this project possible.

We would like to express our special thanks to **Dr. V. Vijaya Kumar**, Dean School of Engineering, Anurag University, for their encouragement and timely support in our B. Tech program.

We would like acknowledge our sincere gratitude for the support extended by **Dr. G. Vishnu Murthy**, Dean, Dept. of CSE, Anurag University. We also express our deep sense of gratitude to **Dr. V V S S S Balaram**, Academic coordinator, **Dr. Pallam Ravi**, Project in-Charge, **Dr. G. Prabhakar Raju**. Project Co-Ordinator and Project review committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stage of our project work.

**LEKKALA SAI KETHANA**  
(20EG105127)

**MADIPADIGE UDAY SAI KIRAN**  
(20EG105128)

**N. MEGHAN SATWIK**  
(20EG105134)

**POTTAPALLY KEERTHIPRIYA**  
(20EG105141)

## **DECLARATION**

We hereby declare that the Report entitled “**MULTI-AUTHORITY ATTRIBUTE BASED KEYWORD SEARCH OVER ENCRYPTED CLOUD DATA**” submitted for the award of Bachelor of Technology Degree is my original work and the Report has not formed the basis for the award of any degree, diploma, associate ship or fellowship of similar other titles. It has not been submitted to any other University or Institution for the award of any degree or diploma.

**LEKKALA SAI KETHANA**  
**(20EG105127)**

**MADIPADIGE UDAY SAI KIRAN**  
**(20EG105128)**

**N. MEGHAN SATWIK**  
**(20EG105134)**

**POTTAPALLY KEERTHIPRIYA**  
**(20EG105141)**

**Place: HYDERABAD**

**Date:**

## ABSTRACT

The proliferation of cloud computing has ushered in a new era of data storage and processing, offering unparalleled scalability and convenience. However, concerns over data privacy and security remain paramount, particularly when sensitive information is entrusted to third-party cloud providers. To mitigate these concerns, encryption techniques have been widely adopted to protect data confidentiality. Yet, traditional encryption methods hinder efficient search and retrieval operations over encrypted data. In response, attribute-based encryption (ABE) has emerged as a promising solution, enabling fine-grained access control based on user attributes. However, in decentralized environments with multiple authorities, traditional ABE falls short. Multi-authority attribute-based encryption (MA-ABE) addresses this gap by allowing multiple independent authorities to define and enforce access policies. This paper explores the intersection of MA-ABE and keyword search encryption, presenting an in-depth analysis of techniques to enable secure and efficient keyword search over encrypted cloud data.

## LIST OF FIGURES

Figure No.	Figure Name	Page No.
3.1	System architecture design	15
3.2	Use case diagram	18
3.3	Patient sequence diagram	19
3.4	Doctor sequence diagram	19
3.5	TA and CS sequence diagram	20
3.6	Class diagram	20
3.7	Working	21
5.1	Home Page	29
5.2	Generation of public and master keys by trusted authority	30
5.3	Generation of secret keys for doctors	31
5.4	Secret key generation using attributes	32
5.5	Patient uploads data to specific doctor	33
5.6	Data files are encrypted using access control policy	34
5.7	Generating symmetric key using keyword	35
5.8	Encryption of keyword using symmetric key	36
5.9	Patient sharing the data files with the doctor	37
5.10	Doctors has a list of keywords shared with them which cannot be viewed by others	38
5.11	Doctor searches patient's data using keywords	39
5.12	Cloud server search the keyword in the database and return the matched data to the doctor	40
5.13	If the access policy satisfies, the doctor can view the encrypted data file	41

5.14	Using secret key doctor can decrypt the encrypted data file	42
5.15	Doctor can download the data file now	43

## LIST OF TABLES

Table No.	Table Name	Page No.
2.1.1	Comparison of Existing Methods	11

## INDEX

S. No.	CONTENT	Page No.
1.	Introduction	9
2.	Literature Review	11
3.	Proposed Method	14
	3.1 Problem Identification	14
	3.1.1 Existing Model	14
	3.2 Proposed System	14
	3.2.1 System Architecture	15
	3.2.2 Entities of the System	15
	3.2.3 Main Contributions of the System	16
	3.3 System UML Diagrams	18
	3.3.1 Use Case Diagram	18
	3.3.2 Sequence Diagram	19
	3.3.3 Class Diagram	20
	3.4 Working	20
4.	Implementation	22
	4.1 Code	22
	4.1.1 Setup	22
	4.1.2 Key Generation	23
	4.1.3 Encryption	23
	4.1.4 Decryption	25
	4.1.5 Trapdoor	25
	4.1.6 View Data	26
5.	Experiment Results	28
	5.1. Experiment Setup	28
	5.2. Experiment Screenshots	29
6.	Discussion Of Results	44
7.	Conclusion	45
8.	Future Scope	46
	References	47



## 1. INTRODUCTION

Cloud computing, a potent technology leveraging the Internet and remote servers, handles vast-scale data maintenance and intricate computations. One significant application lies in personal health record systems, granting individuals access to, management of, and sharing of their health information. Each patient retains full control over their personal health record, enabling sharing with various users, including healthcare providers' staff. To curb storage and operational expenses, numerous large organizations and individual users entrust their personal health records to the cloud, albeit relinquishing some control in the process. However, this reliance on semi-trusted cloud servers exposes personal health records to potential unauthorized access or commercial exploitation. Encrypting personal health records before cloud outsourcing becomes imperative to safeguard confidentiality, yet this hinders conventional search algorithms from operating in the encrypted domain.

Searchable encryption (SE) emerges as a cryptographic solution facilitating specific information retrieval, such as keywords, from encrypted documents without revealing plaintext details. The process involves the data owner encrypting both documents and keywords, then uploading the encrypted data and keyword ciphertext to the cloud. When a data user seeks document retrieval, they generate a keyword token sent to the cloud, which employs a search algorithm to match the keyword token with corresponding keyword ciphertext, returning encrypted documents with matching keywords. Traditional searchable encryption models grant either complete or no access to shared data based on possession of the secret key. However, many data owners desire more nuanced sharing capabilities for their data.

Attribute-based encryption (ABE) provides a solution to the aforementioned issue. ABE involves encrypting files based on attributes associated with each user. Sahai and Waters introduced ABE, which allows for access control over encrypted files through the use of access policies. Specifically, Ciphertext-policy ABE (CP-ABE) is proposed within this framework. In CP-ABE, each ciphertext is linked to a set of attributes, and each user's private key corresponds to an access policy for attributes. Users can decrypt a ciphertext only if the attributes associated with it meet the access policy defined by their private key.

To enable fine-grained access control and keyword search within an e-healthcare cloud computing system, we leverage attribute-based encryption techniques.

Previous research failed to show that current attribute-based methods could simultaneously facilitate keyword search and data sharing within a single scheme. Consequently, there's a need for a secure solution capable of fully supporting both keyword searching and data sharing while ensuring the privacy of keywords. Thus, we propose a ciphertext-policy attribute-based mechanism with keyword search and data sharing (CPAB-KSDS) for encrypted cloud data. This mechanism enables searching and sharing functionalities within a ciphertext-policy framework.

## 2. LITERATURE REVIEW

Author(s)	Method	Advantages	Dis-advantages
D. X. Song, D. Wagner, and A. Perrig	Practical techniques for searches on encrypted data	Pioneering work in the field of searching on encrypted data.  Provides practical techniques for secure searches on encrypted data, which is essential for privacy-preserving information retrieval.	Being an early work, it might lack the sophistication and efficiency of more recent approaches.  It might not address all security and privacy concerns that have emerged since its publication.
D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano	Public key encryption with keyword search	Introduces a novel concept of public key encryption with keyword search, expanding the possibilities of secure search functionality.  Lays the groundwork for further research in this area.	Might have limitations in terms of efficiency and scalability, especially for large-scale deployments.  It might have vulnerabilities or weaknesses that were not identified at the time of publication but have since been discovered.
Q. Zheng, S. Xu, and G. Ateniese	Vabks: verifiable attribute-based keyword search over outsourced encrypted data	Enhances the security and accountability of searches.  Provides mechanisms for users to verify the correctness of search results without revealing sensitive information.  Addresses the issue of outsourcing encrypted data while ensuring search functionality and data integrity.	May involve additional computational overhead for verification processes, potentially impacting the efficiency of search operations.  Implementation complexity might be a challenge for practical deployment in resource-constrained environments.
B. Waters	Ciphertext-policy attribute-based encryption: An expressive,	Offers expressiveness, efficiency, and provable security guarantees.	Complexity in managing attribute-based access policies and associated cryptographic keys, which

	efficient, and provably secure realization	<p>Allows fine-grained access control based on attributes, enhancing the flexibility and usability of encryption schemes.</p> <p>Provides theoretical foundations and practical realizations for attribute-based encryption, addressing the needs of various access control scenarios.</p>	<p>may require careful design and management.</p> <p>Potential overheads in terms of encryption and decryption operations, especially for large-scale deployments or complex access policies.</p>
Baojiang Cui, Zheli Liu	Key-Aggregate Searchable Encryption (KASE) for Group Data Sharing via Cloud Storage	<p>Facilitates efficient and secure data sharing within groups via cloud storage.</p> <p>Enhances privacy and security by enabling encrypted search functionalities while preserving access control and confidentiality.</p> <p>Addresses the need for scalable and flexible solutions for group data sharing in cloud environments.</p>	<p>Complexity in managing and updating keys, especially in large and dynamic group settings, which may introduce overheads and management challenges.</p> <p>Potential vulnerabilities in the encryption scheme or key management mechanisms, which could compromise the security of shared data</p>
Yinbin Miao, Robert H. Deng, Ximeng Liu, Kim-Kwang Raymond Choo, Hongjun Wu, and Hongwei Li	Multi-authority Attribute-Based Keyword Search over Encrypted Cloud Data	<p>Enhances the flexibility and scalability of access control in cloud environments.</p> <p>Allows for fine-grained access control policies based on attributes while enabling keyword search functionalities over encrypted data.</p> <p>Addresses the need for secure and efficient access control mechanisms in</p>	<p>Complexity in managing multiple authorities and associated attributes, which may introduce challenges in policy enforcement and key management.</p> <p>Potential performance overheads in terms of encryption, decryption, and search operations, especially in scenarios with complex access policies and large datasets.</p>

		distributed cloud environments with multiple authorities.	
V. Goyal, O. Pandey, A. Sahai, and B. Waters	Attribute-based encryption for fine-grained access control of encrypted data	<p>Addresses the need for fine-grained access control of encrypted data, which is essential for maintaining privacy and security in diverse applications.</p> <p>Offers attribute-based encryption techniques, providing more flexibility in access control policies.</p>	Complexity in managing attributes and access policies, which may require careful design and implementation.
J. Li, X. Lin, Y. Zhang, and J. Han	Ksf-oabe: outsourced attribute-based encryption with keyword search function for cloud storage	<p>Introduces a novel approach for outsourced attribute-based encryption with keyword search functionality, addressing the need for efficient and secure data retrieval in cloud storage.</p> <p>Enables users to securely search for specific data based on attributes while preserving confidentiality.</p> <p>Facilitates outsourcing of encryption and search operations to reduce computational overheads on the client side.</p>	<p>May introduce complexities in managing attribute-based access policies and associated cryptographic keys, potentially increasing the risk of misconfigurations or security vulnerabilities.</p> <p>Efficiency and scalability of the scheme may vary depending on factors such as the size of the dataset, the complexity of access policies, and the computational capabilities of the cloud infrastructure.</p>

Table 2.1.1 Comparison of Existing Methods

### **3- PROPOSED METHOD**

#### **3.1 Problem Identification**

In the current landscape of cloud computing, both organizations and individuals are increasingly reliant on remote storage solutions to handle extensive volumes of sensitive data. However, ensuring the privacy and confidentiality of this data presents notable hurdles, particularly when striving to achieve efficient search capabilities over encrypted data, all while adhering to multi-authority access control policies.

The principal challenge tackled by this project involves creating a resilient system capable of facilitating secure keyword searches over encrypted cloud data while accommodating the demands of multi-authority attribute-based access control. This entails the development of mechanisms that empower authorized users to search for specific information based on keywords or attributes, while concurrently preventing unauthorized access attempts to sensitive cloud-stored data.

##### **3.1.1 Existing Model**

Most of the existing schemes do not support keyword searching schemes. In such cases, data users must download, filter and process a large amount of data in order to get relevant results, which obviously lack practicality. In the existing system, there is a not supporting search technique on outsourced encrypted data. Even not authorized people are also getting the cloud-encrypted results. Due to this issue, increases the computational cost of providing cloud data.

#### **3.2 Proposed System**

The proposed system allows data owners to control the search permission for their outsourced encrypted data according to an access control policy. As long as his attributes satisfy the access control policy, any user can perform a keyword search. This means that our primitive supports multiuser search. In addition, every user with a set of attributes can generate a delegated key for another user who has a more restricted set of attributes.

### 3.2.1 System Architecture:

The proposed system model consists of four components namely Data Owner (patients), Data User (doctors belonging to different hospitals), Trusted Authority and Cloud Service Provider.

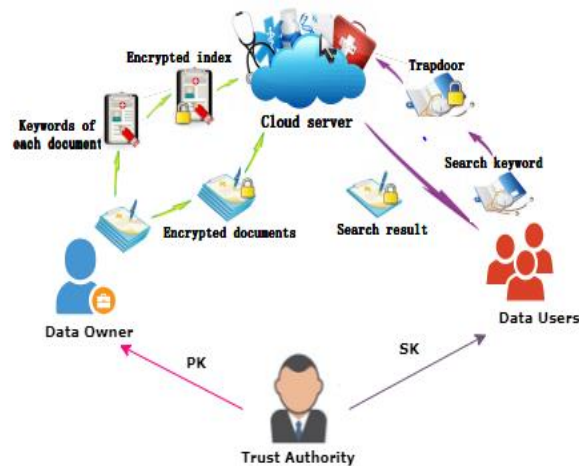


Figure 3.1 : System architecture design

### 3.2.2 Entities of the System:

#### Data Owners:

Data owners are entities or individuals who possess ownership rights over the data stored in the cloud. They are responsible for determining access control policies, defining attributes, and specifying which users or authorities have permission to access specific data based on their attributes.

Data owners collaborate with trusted authorities to establish and enforce access control policies that govern data access within the cloud environment. They may also interact with data users to grant access to relevant data based on predefined attributes and search criteria.

#### Data Users:

Data users are individuals or entities who require access to specific data stored in the cloud for various purposes, such as information retrieval, analysis, or decision-making.

Data users submit search queries containing keywords or attributes to retrieve relevant encrypted data from the cloud storage while preserving data privacy.

They rely on the mechanisms established by the trusted authorities and data owners to access encrypted data securely and efficiently based on their authorization levels and attributes.

#### **Trusted Authority:**

Trusted authorities are entities responsible for managing and enforcing access control policies in the multi-authority attribute-based system. They collaborate with data owners to define attribute-based access control policies and ensure that these policies are enforced consistently across the cloud environment.

Trusted authorities authenticate users, validate their access rights based on attributes, and facilitate secure keyword search operations over encrypted data. They play a central role in mediating interactions between data owners, data users, and the cloud service provider to ensure secure and efficient data access and retrieval.

#### **Cloud Service Provider:**

The cloud service provider hosts and manages the infrastructure and resources required for storing and processing encrypted data on behalf of data owners and users. The provider ensures the confidentiality, integrity, and availability of the data stored in the cloud environment and adheres to service-level agreements (SLAs) regarding data security and privacy.

While the cloud service provider does not have access to the plaintext data, it facilitates encrypted data storage, retrieval, and processing operations as per the instructions and policies defined by data owners, users, and trusted authorities.

### **3.2.3 Main Contributions of the System:**

#### **a. Attribute-Based Encryption (ABE):**

Attribute-based encryption (ABE) is a cryptographic method utilized for enforcing access control policies reliant on attributes. In this context, the patient employs ABE to encrypt data before transferring it to the cloud. Each user is linked to a set of attributes, and access policies are formulated accordingly. ABE ensures that data is encrypted to allow decryption solely by users possessing specific attributes.



**b. Multi-Authority Setup:**

In this scenario, multiple authorities may exist, each having its own attribute authorities tasked with overseeing access control policies. Every authority has the autonomy to establish its unique set of attributes and access policies for its data. The implementation of a multi-authority setup guarantees decentralized management of attributes and access policies, thereby improving scalability and flexibility.

**c. Keyword Searchable Encryption:**

Conventional ABE schemes lack the capability for keyword search within encrypted data. Nonetheless, in this context, users require the ability to search for particular keywords within encrypted documents. To achieve this, techniques like searchable encryption are utilized, preserving data confidentiality while enabling keyword searches. Approaches such as attribute-based keyword search (ABKS) empower authorized users to search for keywords based on their attributes, all without disclosing document contents to the cloud server.

**d. Secure Index Generation:**

To facilitate keyword search within encrypted data, secure indexes are created for individual documents based on their contained keywords. These indexes undergo encryption using ABE, guaranteeing access solely to users possessing relevant attributes. The generation of secure indexes encompasses methods like constructing inverted indexes and employing ABE for encryption.

**e. Fine-grained Keyword Search Operation:**

When a user intends to search for a particular keyword, they send a search query to the cloud server. Equipped with the requisite cryptographic keys and access policies, the cloud server conducts the search operation on the encrypted data and provides the relevant results to the user. The search process is structured to ensure that the cloud server gains no knowledge of the plaintext data or the search query.

**f. Access Control Mechanism:**

Prior to allowing access to search outcomes, the cloud server confirms that the attributes of the requesting user align with the access policies linked to the encrypted data. Through access control enforcement, only authorized users possessing pertinent attributes are permitted to access the search results.

**g. Privacy and Security Considerations:**

Throughout the procedure, several security and privacy concerns need attention, encompassing safeguarding data confidentiality, guaranteeing data integrity, thwarting unauthorized access, and managing risks like insider attacks and collusion.

### 3.3 System UML Diagrams

#### 3.3.1 Use Case Diagram

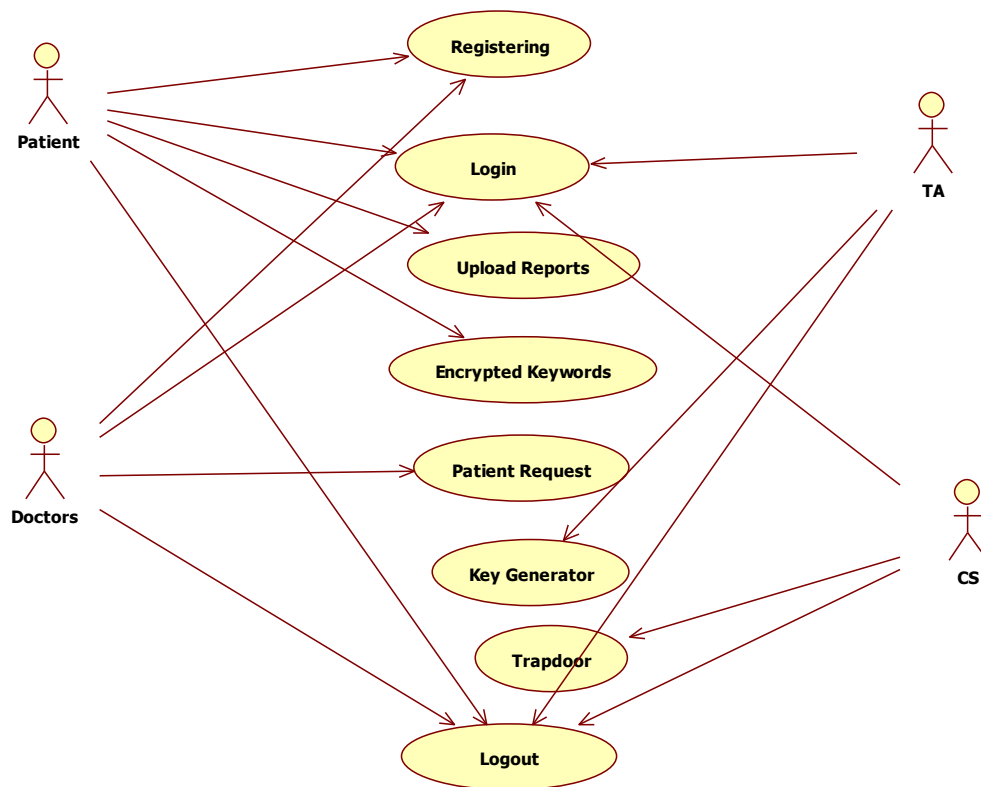


Fig 3.2 Use case Diagram

### 3.3.2 Sequence Diagram

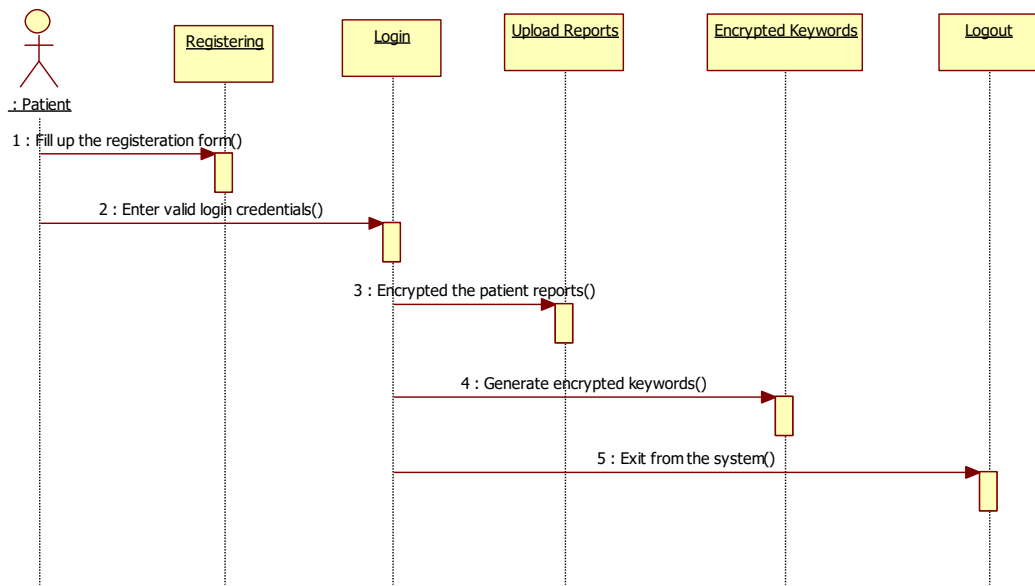


Fig 3.3 Patient Sequence Diagram

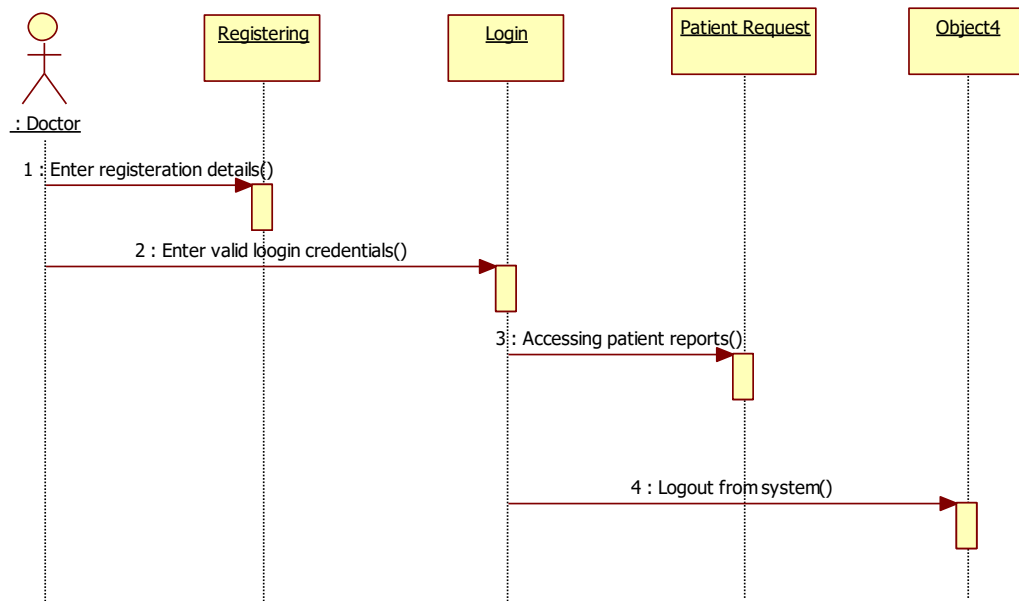


Fig 3.4 Doctor Sequence Diagram

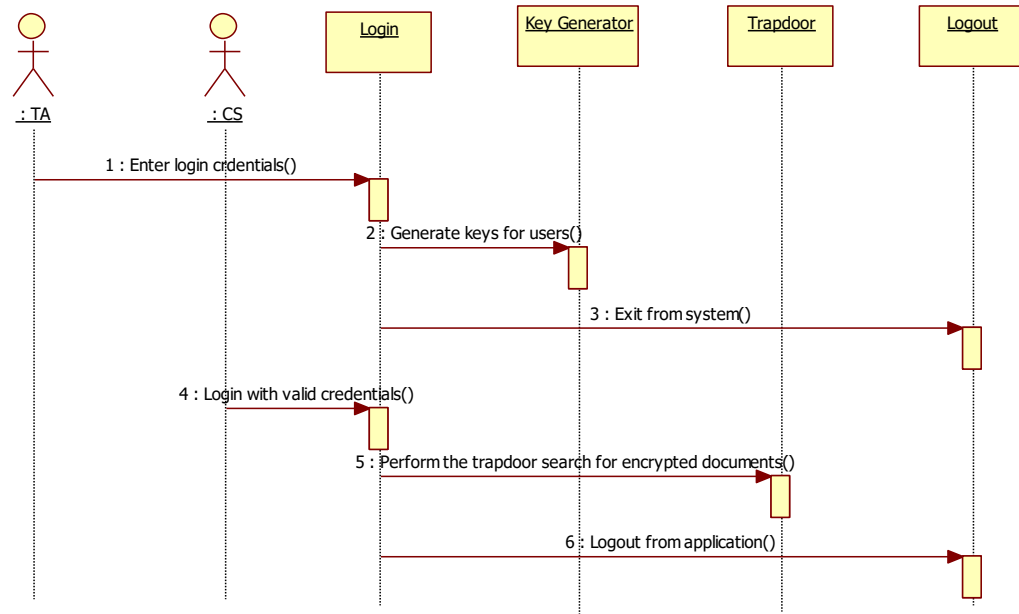


Fig 3.5 Ta and CS sequence Diagram

### 3.3.3 Class Diagram

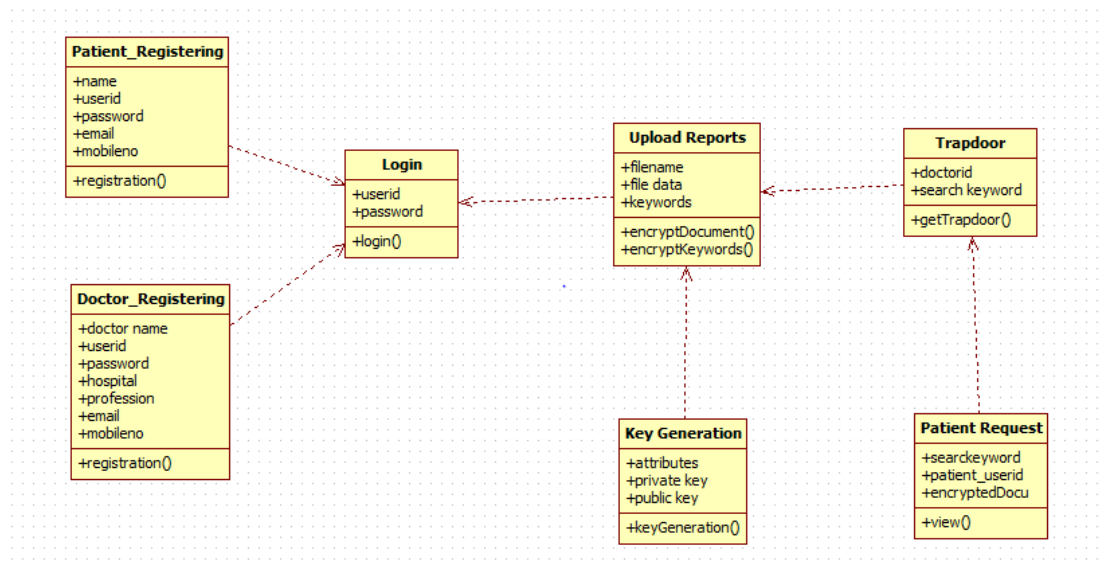


Fig 3.6 Class Diagram

### 3.4 Working:

Initially, the Data Owners and Data Users will register with the system. Trusted Authority will generate public key and master key. Then, by using public key, master key and user attributes (user id, hospital name, specialization) it will also generate private keys (secret key) for each data user i.e., doctors.

Now, the data owners will encrypt the data records using their public key, user attributes and keyword before outsourcing the them. The keyword is also encrypted by using a random symmetric key and will be shared to the specific data owner only. Thus, the data owner will share or upload the encrypted data records to the cloud server ensuring that data remains encrypted even during search operations. This prevents unauthorized access to plaintext data by the cloud service provider or any unauthorized entities.

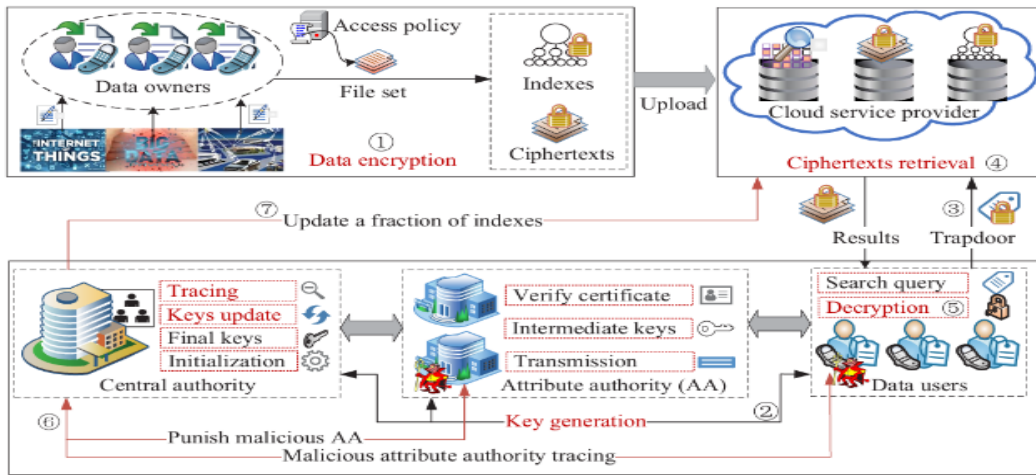


Fig. 3.7 Working

If the data owner i.e., a particular doctor wants to view the data records, he can only able to access the encrypted data records shared to him. Using the keywords which are shared by the data owners, the data user can search for the data records. This moment the Cloud Server searches for the encrypted data records matching the keyword and results the relevant data records only. Thus, efficient search functionality is achieved as the users can retrieve relevant information promptly without compromising data privacy.

Here the encrypted data records can be decrypted and downloaded by the data user using their secret key. If the user's attributes satisfy the access policy, then only they can decrypt and download those encrypted data records with a private key. Otherwise, they are denied to access those data records.

## 4. IMPLEMENTATION

### 4.1 Code:

#### 4.1.1 Setup

```
<%
CPABE abe = new CPABE();
abe.setUp();
Element pk=abe.getPublicKey();
System.out.println("Pk"+pk);
Element mk=abe.getMasterKey();
Connection conl=databasecon.getconnection();
Statement st=conl.createStatement();
ResultSet rst=st.executeQuery("select *from setup");
if(!rst.next()){
    PreparedStatement p=conl.prepareStatement("insert into setup values(?,?)");
    p.setString(1,pk.toString());
    p.setString(2,mk.toString());
    int i=p.executeUpdate();
    if(i>0){
        %>
        <font color="green" size="+2">      <center><strong>PK , MSK are
Generated.</center></strong><br>
        <%}
    }else{%>
        <font color="RED" size="+2"> <center><strong>Keys are already
Generated.</strong></center>

        <%
    }
    %>
```

#### 4.1.2 Key Generation

```
<%  
try{  
    Connection con = databasecon.getConnection();  
    Statement st=con.createStatement();  
    String pk=request.getParameter("pk");  
    String msk=request.getParameter("msk");  
    String atrbt=request.getParameter("atrbt");  
    String atrbts[]=atrbt.split(",");  
    CPABE e=new CPABE();  
    Element SK=e.keygen(pk,msk,atrbts);  
}%>  
  
<%  
    Statement st1=con.createStatement();  
    int i =st1.executeUpdate("update doctor set sk='"+SK.toString()+" where  
doctuid='"+request.getParameter("unm")+"' ");  
    if(i>0){  
        st.executeUpdate("update doctor set status='active' where  
doctuid='"+request.getParameter("unm")+"' ");  
        response.sendRedirect("doctors.jsp?msg=done");  
    }  
}  
catch(Exception e)  
{  
    e.printStackTrace();  
}  
}%>
```

#### 4.1.3 Encryption

```
<%  
    String pk = (String)session.getAttribute("pk");  
    String acesp =(String)session.getAttribute("acesp");  
    //String data = request.getParameter("query");
```

```

        String peid= (String)session.getAttribute("pid");
    try{
        Connection con=databasecon.getconnection();
        Statement stt=con.createStatement();
        ResultSet r12=stt.executeQuery("select *from temp");
        if(r12.next())
        {
            docnm=r12.getString(1);
            data=r12.getBytes(2);
        }
        Statement st=con.createStatement();
        ResultSet r1=st.executeQuery("select max(qid) from cloud");
        if(r1.next())
        {
            qid=r1.getInt(1);
            qid++;
        }
        PreparedStatement ps=con.prepareStatement("INSERT INTO cloud
(qid,puid,data,docnm,aces)VALUES(?,?,"+CPABE_encrypt+"(?,?,?)");
        ps.setInt(1,qid);
        ps.setString(2,peid);
        ps.setBytes(3,data);
        ps.setString(4,docnm);
        ps.setString(5,acesp);
        int s = ps.executeUpdate();
        Connection con1=databasecon.getconnection();
        Statement st1=con1.createStatement();
        ResultSet r11=st1.executeQuery("select data from cloud where qid="+qid+" ");
        if(r11.next())
        {
            ct=r11.getBytes(1);
        }}
        catch(Exception e){e.printStackTrace();}
    %>

```



#### 4.1.4 Decryption

```
<%
String qid=request.getParameter("qid");
String docid= (String)session.getAttribute("docid");
String sk=request.getParameter("sk");
String pid = request.getParameter("pid");
Connection con1 = databasecon.getconnection();
Statement st1 = con1.createStatement();
Statement st=con1.createStatement();
ResultSet r11=st.executeQuery("select
"+CPABE_decrypt+"(data,'" +sk+"'"),docnm from cloud where qid='"+qid+"
");
if(r11.next())
{
data=r11.getBytes(1);
docnm=r11.getString(2);
}
%>
```

#### 4.1.5 Trapdoor

```
try{
Connection con=databasecon.getconnection();
Statement st=con.createStatement();
ResultSet rs=st.executeQuery("select * from trapdoor where status='request'
");
while(rs.next())
{
%>
<tr><td><%=rs.getString("doctuid")%><td><%=rs.getString("keywords")%>
<td><a
href="trapdoor2.jsp?docuid=<%=rs.getString("doctuid")%>&kwrd=<%=rs.getString(
"keywords")%>"><h4>Search</a>
<%
```

```

    }
    } catch (Exception e) {
        System.out.println(e);
    }
}
%>

```

#### 4.1.6 View Data

```

<%
String pass=null,uid=null;
String prof=null,hosp=null,test2=null;
String qid = request.getParameter("qid");
String pid = request.getParameter("puid");
try
{
    Connection con1 = databasecon.getConnection();
    Statement st1 = con1.createStatement();
    String sss1 = "select docprof,dochosp from doctor where
doctuid='"+session.getAttribute("docid")+"' ";
    System.out.println("ss="+sss1);
    ResultSet rs1=st1.executeQuery(sss1);
    if(rs1.next()){
        prof=rs1.getString("docprof");
        hosp=rs1.getString("dochosp");
    }
    String docid=(String)session.getAttribute("docid");
    String aces=prof+"&&" +hosp+"&&" +docid;
    PreparedStatement p=con1.prepareStatement("select *from cloud where qid=?
and aces=?");
    p.setString(1,qid);
    p.setString(2,aces);
    ResultSet rs=p.executeQuery();
    if(rs.next()){
        response.sendRedirect("viewdata2.jsp?qid="+qid+"&pid="+pid);
    }
}
} catch (Exception e) {
    System.out.println(e);
}
%>

```

```
        }else{
            response.sendRedirect("patientReq.jsp?msg=NotAccess");
        }
    }
    catch(Exception e1)
    {
        out.println(e1);
    }
    %>
```

## **5: EXPERIMENTAL RESULTS**

### **5.1 Experimental Setup:**

#### **Java:**

In this system, for developing the business logic of our application, we are using Java technology with 1.8 versions. In this system, we had used the Java language for developing the web application jsp tags. It is most comfortable language to build web-based application.

#### **Visual Studio code IDE:**

Visual Studio is an integrated development environment (IDE) created by Microsoft. It provides a comprehensive set of tools and services for software development, making it easier for developers to create, debug, test, and deploy applications. Visual Studio supports a wide range of programming languages, including C++, C#, Visual Basic, F#, Python, and more.

#### **Apache Tomcat server:**

In this system, we are using Apache Tomcat application server for deploy our web application and running at the browser side. we had used the local tomcat server for running my application at browser side. It is open source, so every one used this application server unlike the glass fish, web logic server and, etc.

#### **MySQL database:**

For storing the application information like patient and doctors details, etc. we are using MySQL database server with 5.5 version. It is the most user-friendly and comfortable tool for working with DBMS.

#### **SQLyog:**

In this system, we had installed the SQLyog GUI application for the MySQL database server. Using this tool, we created the tables easily without typing the SQL commands. With this tool, we had performed all CRUDT operations to support my application.

## 5.2 Experimental Screenshots:



Fig 5.1 Home page

Once we open the Apache Tomcat web server, we will be directed to the home page given Fig 5.1.

Here we can find four entities namely patient, doctor, TA and CS. Respective users need to register and login to the system. The details of data owners, data users and other information will be stored in the database.



**PK , MSK are Generated.**

Fig 5.2 Generation of public and master keys by trusted authority

Now, Trusted Authority will generate public and master keys once we click on setup. These keys will be generated only once in the system.

New Doctors				
DoctorName	Userld	Profession	Hospital	KeyGen
keerthi	keerthi	Cardiologist	Hosp3	GenSK
krishna	krishna	Cardiologist	Hosp2	GenSK

Fig 5.3 Generation of secret keys for doctors

The trusted authority will generate secret keys for every data user i.e., doctor. Just click on doctors to view the list of doctors to generate secret keys and click on GenSK shown in Fig 5.3.

## KEY GENERATION

Public Key ( PK ) :

236539860562925269623405674018065106838878012359

Master key ( MSK ) :

78958393610260396677941180177968919697066428932

Attributes ( S ) :

keerthi,Hosp3,Cardiologist

GetPrivateKey

Fig 5.4 Secret key generation using attributes

Using the public key, master key and user attributes such as user name, hospital name and user profession, secret key will be generated.



## UPLOAD REPORTS

Profession & Hospital

Cardiologist&Hosp3

Select physician

keerthi

Next

Fig 5.5 Patient uploads data to specific doctor

Now if data owner i.e., patient wants to upload his reports to a particular data user i.e., doctor, he should to login in to the system and click on upload reports.

The patient will choose the particular doctor to whom he wants to share the reports which is shown in Fig 5.5. First, the profession and hospital name should be chosen so that the list of corresponding doctors will be shown. From that choose the doctor for sharing reports.

## UPLOAD REPORTS

Public key ( PK )

236539860562925269623405674018065106838878012359

Access Policy ( T )

Cardiologist&&Hosp3&&keerthi

Data ( M )

Choose File Abstract.docx

Encryption

Fig 5.6 Data files are encrypted using access control policy

Using the public key and user attributes such as doctor specialization, hospital name and doctor name, the data file is encrypted which ensures data confidentiality.

## UPLOAD REPORTS

The screenshot shows a web interface titled "UPLOAD REPORTS". It contains two input fields. The first field is labeled "CipherText( CT )" and contains the text "[B@1de3f7ee". The second field is labeled "Keywords" and contains the text "reports". Below these fields is a blue button with the text "Get SymmetricKey".

Fig 5.7 Generating symmetric key using keyword

Here the patient should give a keyword to his encrypted data file so that by using this keyword only, the doctor can search and retrieve the respective encrypted data file. This keyword will be shared to the particular doctor only so that other users cannot access the encrypted data file.

Using the encrypted file name and keyword, a symmetric key will be generated. Symmetric key is a random number.

## UPLOAD REPORTS

CipherText( CT )

[B@781e1555

Keywords

reports

SymmetricKey

880751

EncryptionKW

Fig 5.8 Encryption of keyword using symmetric key

By using the encrypted file name, keyword and symmetric key, the keyword will be encrypted.

## UPLOAD REPORTS

CipherText( CT )

{B@175c9392

Keywords :

y?i]?cY)O%wXORE

Share

Fig 5.9 Patient sharing the data files with the doctor

The keyword will be encrypted and sent to the particular doctor so that by using this keyword only, the doctor can search and retrieve the respective encrypted data file. This keyword will be shared to the particular doctor only so that other users cannot access the encrypted data file.


<div> <a href="#">Home</a> <a href="#">Patient Request</a> <a href="#">Shared Keywords</a> <a href="#">Logout</a> </div>		
		
RequestId	Reports	Keywords
10	p5.jpg	data
12	p4.jpg	surgery

Fig 5.10 Doctors has a list of keywords shared with them which cannot be viewed by others

Now if any doctor wants to access particular patient data, first he should to login in to the system. He should to click on patient request to view patient data. There he need to give the keyword to search the encrypted data file from the cloud. Only if keyword matches the encrypted data file will be retrieved.

The doctor can view the list of keywords shared with him by clicking shared keywords in the doctor login page.

A search interface. It consists of a white rectangular box with a thin grey border. Inside the box, the word 'reports' is written in a simple, dark font. Below the text input area is a solid blue button with the word 'Search' in white, rounded corners.

Fig 5.11 Doctor searches patient's data using keywords

The doctor should give the keyword so that the corresponding encrypted data file will be searched in the cloud.

<a href="#">Home</a> <a href="#">Trapdoor</a> <a href="#">Logout</a>		
Trapdoor		
Doctorid	Keywords	Action
kearthi	reports	<a href="#">Search</a>

Fig 5.12 Cloud server search the keyword in the database and return the matched data to the doctor

The cloud server will search for the files which matches the given keyword and returns the files to the doctor.



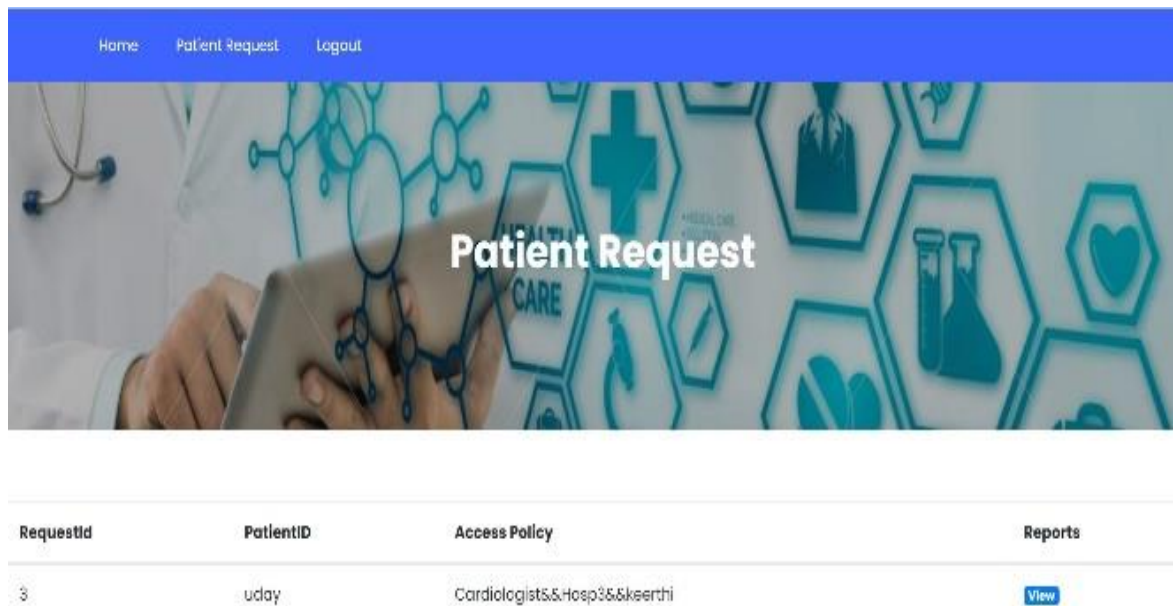


Fig 5.13 If the access policy satisfies, the doctor can view the encrypted data file

The doctor gets the files which are matched with the given keyword. Once he click on view shown in Fig 5.13, the file can be accessible to the doctor if and only if the access policy satisfies i.e., the user attributes matches. If access policy is not satisfied, it will show that ‘the data cannot be viewed’.

## VIEW REPORTS

File Name

Abstract.docx

CipherText( CT )

|B@74e10b8e

Secret key ( SK )

280898739435164044032151855240143064578581324343

Decrypt

Fig 5.14 Using secret key doctor can decrypt the encrypted data file

If the access policy matches, then by using the secret key the doctor can decrypt the encrypted file. This ensures data privacy and confidentiality.

## VIEW REPORTS

File Name

Abstract.docx

File Data

[B@68c967f3

Download

Fig 5.15 Doctor can download the data file now

Now if the secret key is valid one, the doctor can download the patient's data file.

## **6: DISCUSSION OF RESULTS**

The primary objective of the system is to furnish an optimal user experience for hospital personnel, enabling efficient searching of patient records while adhering to access control policies. Before transferring data to the cloud, the patient encrypts it using attribute-based encryption (ABE), ensuring that only users who are equipped with specific attributes can decrypt and access it. Access policies, based on users' attributes, provide precise control over access to patient records.

Authorized users can send search queries to the cloud server, specifying keywords related to patient conditions, treatments, or demographics. The cloud server conducts search operations on the encrypted indexes, identifying relevant documents based on the search criteria. Only documents meeting the access control policies of the querying user are returned as search results, ensuring data confidentiality and compliance with privacy regulations.

The system employs cryptographic methods to safeguard patient data confidentiality and prevent unauthorized access. Robust access control mechanisms enforce granular access policies, reducing the risk of data breaches and insider threats. Measures to preserve privacy are implemented to prevent the disclosure of sensitive information during search operations or access control verifications.

## 7: CONCLUSION

The system is designed to accommodate multiple authorities, thereby avoiding performance bottlenecks typically associated with centralized systems in cloud environments. As the data will be encrypted before it is outsourced, integrity and confidentiality of the data is maintained. Also, only relevant search results are retrieved using keywords which significantly boosts the systems overall performance. They are provably secure, ensuring that the untrusted server gains no insight into plaintext from ciphertext alone. They uphold query isolation, preventing the server from extracting additional information beyond search results. Controlled searching restricts the server from querying arbitrary words without user authorization.

Additionally, the introduced MABKS system enables the tracing of malicious Attribute Authorities (AAs) to mitigate collusion attacks and supports attribute updates to prevent unauthorized access using outdated secret keys.

However, a notable limitation is that the MABKS system does not support advanced search queries such as conjunctive keyword search, fuzzy search, or subset search. Future efforts will concentrate on developing an efficient and adaptable index construction method to enable the MABKS system to handle a variety of search requests effectively.

## **8: FUTURE SCOPE**

In the realm of Multi-authority Attribute-Based Keyword Search over Encrypted Cloud Data (MA-ABKS), the future holds promising avenues for advancement. One key area of exploration lies in the development of more sophisticated search capabilities to accommodate diverse user needs. By enhancing the system's capacity to support advanced queries such as conjunctive keyword search, fuzzy search, and subset search, researchers can broaden its applicability across various domains. This expansion would enable users to extract more nuanced insights from encrypted data stored in the cloud, fostering greater efficiency and effectiveness in information retrieval tasks. Moreover, advancements in search techniques could facilitate the discovery of complex patterns and relationships within encrypted datasets, unlocking new opportunities for data-driven decision-making and knowledge discovery.

Another vital direction for future research involves the refinement of privacy-preserving mechanisms to bolster user confidentiality and data security. As privacy concerns continue to escalate, there is a growing imperative to develop robust safeguards against unauthorized access and data breaches. By exploring innovative encryption schemes, authentication protocols, and privacy-preserving techniques, researchers can fortify the MA-ABKS system's defenses against emerging threats. Additionally, efforts to enhance interoperability with existing systems and adherence to industry standards will be critical for ensuring seamless integration into healthcare IT infrastructure. Through these endeavors, the MA-ABKS project can pave the way for a more secure, efficient, and privacy-preserving approach to keyword search over encrypted cloud data, ultimately empowering healthcare organizations to leverage the full potential of their data assets while safeguarding patient privacy.

## REFERENCES

- [1] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. IEEE Symposium on Security and Privacy (SP'00), 2000, pp. 44–55.
- [2] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Proc. Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'04), vol. 3027, 2004, pp. 506–522.
- [3] Q. Zheng, S. Xu, and G. Ateniese, "Vabks: verifiable attribute-based keyword search over outsourced encrypted data," in Proc. IEEE Conference on Computer Communications (INFOCOM'14), 2014, pp. 522–530.
- [4] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in Proc. International Conference on Practice and Theory in Public Key Cryptography (PKC'11), vol. 6571, 2011, pp. 53–70.
- [5] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," IEEE Transactions on parallel and distributed systems, vol. 25, no. 1, pp. 222–233, 2014.
- [6] Baojiang Cui, Zheli Liu, "Key-Aggregate Searchable Encryption (KASE) for Group Data Sharing via Cloud Storage", IEEE Transactions on Computers, January 2015.
- [7] H. Li, D. Liu, Y. Dai, T. H. Luan, and X. S. Shen, "Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage," IEEE Transactions on Emerging Topics in Computing, vol. 3, no. 1, pp. 127–138, 2015.
- [8] Yinbin Miao, Robert H. Deng, Fellow, IEEE, Ximeng Liu, Kim-Kwang Raymond Choo, Senior Member, IEEE, Hongjun Wu, and Hongwei Li "Multi-authority Attribute-Based Keyword Search over Encrypted Cloud Data" IEEE Transactions on Dependable and Secure Computing (Volume: 18, Issue: 4 01 July-Aug. 2021)
- [9] Sahai, A. and Waters, B. (2005) Fuzzy Identity-Based Encryption. In Annu. Int. Conf. the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, pp. 457–473, Springer, Berlin.
- [10] X. Mao, J. Lai, Q. Mei, K. Chen, and J. Weng, "Generic and efficient constructions of attribute-based encryption with verifiable outsourced decryption," IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 5, pp. 533–546, 2016.
- [11] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. ACM conference on Computer and communications security (CCS'06), 2006, pp. 89–98.

- [12] J. Li, X. Lin, Y. Zhang, and J. Han, "Ksf-oabe: outsourced attribute-based encryption with keyword search function for cloud storage," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 715–725, 2017.
- [13] Y. Miao, X. Liu, K.-K. R. Choo, R. H. Deng, J. Li, H. Li, and J. Ma, "Privacy-preserving attribute-based keyword search in shared multi-owner setting," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, pp. 1–15, 2019.